Towards Plug-n-Play robot guidance: Advanced 3D estimation and pose estimation in Robotic applications

Sølund, Thomas

Publication date:
2017

Document Version
Publisher's PDF, also known as Version of record

Towards Plug-n-Play robot guidance: Advanced 3D estimation and pose estimation in Robotic applications

Thomas Sølund

Kongens Lyngby 2016

Technical University of Denmark
Department of Applied Mathematics and Computer Science
Richard Petersens Plads, building 324,
2800 Kongens Lyngby, Denmark
Phone +45 4525 3031
compute@compute.dtu.dk
www.compute.dtu.dk

Summary (English)

Robots are a key technology in the quest for higher productivity in Denmark and Europe. Robots have existed in many years as a part of production lines where they have solved monotonous and repetitive task in mass production industries. Typical the programming of these robots are handled by engineers with special knowledge who have often raised the price for using robots to a given production task. If robots have to be applicable for small and medium sized enterprises where production task often changes and batch sizes are below 50 products it is necessary that the staff is capable of re-programming the robot by themselves.

During the last five years a number of collaborative robots are introduced on the marked e.g. Universal Robot, which enables a production worker to program the robot to solve simple tasks. With the collaborative robot the production worker is able to make the robot grind, mill, weld and move objects, which are physical located at the same positions. In order to place objects in the same position each time, custom-made mechanical fixtures and aligners are constructed to ensure that objects are not moving. It is expensive to design and build these fixtures and it is difficult to quickly change to a novel task. In some cases where objects are placed in bins and boxes it is not possible to position the objects in the same location each time.

To avoid designing expensive mechanical solutions and to be able to pick objects from boxes and bins, a sensor is necessary to guide the robot. Today, primarily 2D vision systems are applied in industrial robotics, which are in-flexible and hard to program for the production workers. Smart cameras, which are easier

to re-configure and program to detect objects exist. However, computing the correct position such that a robot can move to this position is still a challenge which requires calibration processes. Moreover, the ability to make the solution robust such that it is running 24/7 in a production is demanding and requires the right skills. Basically, the vision part of a flexible automation solution is difficult to manage for a production worker while the robot motion programming is easily handled with the new collaborative robots. This thesis deals with robot vision technologies and how these are made easier for production workers program in order to get robots to recognize and compute the position of objects in the industry.

This thesis investigates and discusses methods to encapsulate a 2D vision system into a framework in order to make changes in production task easier. The framework is presented in [Contribution B] and [Contribution C] and demonstrates how re-configuration of vision systems is made easier but in the same time reviles some of the fundamental problems that exist by observing a tree dimensional world through a two dimensional vision system. This requires a calibration procedure every time in order to convert 2D to 3D, which still is a cumbersome process for a production worker.

For this reason, the rest of the thesis investigates and discusses how 3D computer vision techniques can ease the problem of recognizing and computing the position of objects. In [Contribution D] a small lightweight 3D sensor is presented. The 3D sensor has a size that makes it suitable for tool mounting at a collaborative robot. It is based on structured light principles and 3D estimation techniques, which enables fast and accurate acquisition of point clouds of low textured and reflective industrial objects.

In [Contribution E] a 3D vision system for easy learning of 3D models is presented. The system creates a 3D model of the object by scanning it from three views. Then the object acts as a reference model in the system when new instances of the object have to be located in the scene. With this approach fast re-configuration is possible. In [Contribution F] a new dataset for 3D object recognition and an evaluation of state-of-the-art local features for object recognition are presented. The contribution shows as expected that state-of-the-art 3D object recognition algorithms are not good enough to locate industrial objects with few local shape features on the surface.

Summary (Danish)

Robotter er en nøgleteknologi i søgen efter at øge produktiviten i Danmark og Europa. Robotter har eksisteret i mange år, hvor de har indgået i produktlinjer og løst ensidige og gentagende opgaver. Programmeringen af robotter er blevet varetaget af ingeniører, der traditionelt har øget prisen for at få robotter til at løse nye produktionsopgaver. Hvis robotter skal benyttes af små og mellemstore virksomheder, hvor opgaverne ofte skifter og ordrestørrelsen er mindre end 50 produkter, er det nødvendig at medarbejdere selv kan om-programmere robotterne.

I løbet af de sidste 5 år er der blevet introduceret flere nye samarbejdende robotter som f.eks. Universal Robot der gør det muligt for en produktionsmedarbejder selv at programmere robotten til at løse simple opgaver. Med en samarbejdende robot kan en produktionsmedarbejder få robotten til at slibe, fræse, svejse og flytte emner der ligger fysisk det samme sted. For at emner kan placeres nøjagtig på samme position benyttes der i dag specialbygget mekanik der sørger for at emnet ikke flytter sig. Dette er bekostlig og gør det besværligt at skifte mellem opgaver. I nogle tilfælde er det ikke muligt at emner er placeret på samme sted bl.a. når emner er placeret i kasser og paller.

For at undgå at lave dyre mekaniske løsninger og være i stand til at tage emner der er placeret i kasser og paller, er en sensor nødvendig for at guide robotten. I dag findes der primært 2D-vision løsninger til robotter, som ikke er fleksible og svære at programmere for en produktionsmedarbejder. Såkaldte "smartkameraer"indeholder funktionalitet der gør det nemmere at få kameraet til at detektere emner. Det er dog stadig en stor udfordring at omregne resultatet til en position, som en robot kan forstå og dermed gøre hele løsningen robust så den

kan fungere i en produktion. Man kan kort sagt sig at de samarbejdende robotter gør det nemt at programmere robottens bevægelser, hvorimod sensordelen af f.eks. en håndteringsproces er sværere at håndtere for en produktionsmedarbejder med den eksisterende teknologi. Denne afhandling omhandler hvordan robot vision teknologi kan gøre det nemmere for en produktionsmedarbejder at få robotter til at genkende og beregne positioner af emner.

Afhandlingen undersøger og diskuterer metoder, der pakker 2D robot vision ind i et samlet framework, der gør det nemt for en produktionsmedarbejder at skifte arbejdsopgave. Frameworket, der præsenteres i [Bidrag B,C], gør det nemmere at omkonfigurere visionsystemet, men viser også det fundamentale problem ved, at betragte verden i to dimensioner. Når der udelukkende benyttes 2D computer vision i en tredimensionel verden skal visionsystemet kalibreres hver gang, således at den todimensionelle verden set igennem et kamera kan omsættes til tre dimensioner.

Af samme årsag undersøges og diskuteres i resten af afhandlingen, hvorledes 3D computer-vision-teknikker kan afhjælpe overstående problemstilling. I [Bidrag D] præsenteres en lille letvægts-3D-sensor der har en størrelse, som gør at sensoren kan monteres på en samarbejdende robot. 3D sensoren er baseret på struktureret lys og 3D-estimeringsteknikker, der gør det muligt hurtigt at lave nøjagtige 3D punkt skyer af gængse industrielle emner der typisk kun har lidt tekstur og har skinnende overflader.

I [Bidrag E] præsenteres et 3D-vision-system, der gør det muligt for en produktionsmedarbejder at lave en 3D-model af et emne ved at placere det på et bord, hvorefter modellen benyttes som referenceemne, når andre emner af samme type skal lokaliseres. Systemet muliggør hurtig træning af nye emner. I [Bidrag F] præsenteres et nyt 3D-objektgenkendelses datasæt og en evaluering af state-of-the-art lokale features til 3D-objektgenkendelse. Bidraget viser som forventet at state-of-the-art 3D-objektgenkendelses algoritmer ikke er gode nok når industrielle emner med få lokale features skal genkendes.

Preface

The work presented in this dissertation is the results of three years of study in fulfilment of the requirements for acquiring an industrial Ph.D. degree in computer science funded by The Innovation Fund, Grand Nr: 11-117524. The dissertation is a result of a rewarding co-operation between Danish Technological Institute - Center for Robot Technology and Technical University of Denmark - Department of Applied Mathematics and Computer Science (DTU Compute). The work is done in accordance with the programme of the PhD School (IT-MAN) at DTU Compute for acquiring the PhD degree.

Part of the scientific work in this dissertation was conducted in collaboration with the SDURobotics group at the University of Southern Denmark during an external stay between September 2013 and January 2014. The collaboration established during the external stay at University of Southern Denmark has continued in the rest of the project. The Industrial Ph.D was supervised by my principal supervisor Associated Professor Henrik Aanæs, DTU Compute, Co-supervisor Anders Billesø Besk, DTI. In 2015, Professor Norbert Krüger became a third part supervisor in the project to strengthen the collaboration with University of Southern Denmark.

Odense, 29-July-2016

Thomas Sølund

Acknowledgements

I would like to thank my supervisors, Associated Professor Henrik Aanæs, Teamleader Anders Billesø Beck and Professor Norbert Krüger for supervision and support.

I would also like to thank my company Danish Technological Institute - Center for Robot Technology, for believing in me and supporting my wish to research in 3D Vison despite the fact that 2D vision is still overrepresented in commercial robot solution. Furthermore, I would like to thanks all my great colleagues at DTI both the current and the ones that left the company. Especially, thanks to the Sensor & Process team for supporting my work. At last, I would like to thank the former Director at DTI, Claus Risager for given me the opportunity to apply for a Industrial Ph.D.

Further, I would like to thanks Assistant Professor Anders Glent Buch form University of Southern Denmark for the many productive and rewarding conversations during my Ph.D. As a Ph.D student it is enriching to have a person that understand the difficulties and challenges during a Ph.d. study and is able to explain difficult topics in a good and sometimes easier way than conventional papers and books. Also, thanks to my fellow Ph.D students at DTU Compute for inputs and technical support.

Finally, I would like to thank my wife, Maria and my two daughters Astrid and Martha, for being very supportive and patient with me, while I worked on my Ph.D. Without you I could not have done it. Thanks!

Contents

Chapter 1

Introduction

In the factory of the future many production tasks will be conducted by autonomous robots in collaboration with humans. With the introduction of the industry 4.0 strategy that outlines the future of production and robot technology, computer vision technologies will be a central part in completing this strategy. Industry 4.0 is considered as the fourth industrial revolution, which is based on cyper-physical production systems (CPPS) and will enable a data-driven and agile production in smart-factories in the western world [KWH13]. One of the key elements in cyber-physical systems is to monitor and detect the physical processes in the production and from this information make decentralized decisions and trigger actions. Moreover, industrial production systems have to communicate by machine-to-machine communication and cooperate with humans and each other in real time.
With the third industrial revolution, which started in late 1960s, the use of electronic and information technologies was introduced to further automate productions, especially with the introduction of industrial robots. In the coming decades more industrial robots were introduced at the factory floors to make mass production. In the transformation from the third to the fourth industrial revolution, which is currently taking place, data-driven productions are inevitable to accommodate the requirements to agile production facilities which are able to customize products. In such a production environment, industrial production systems must be able to adapt the production to very short batches or even down to individual products. Further, the production machinery must

automatically adapt the production machinery to handle new requirements and conditions in an agile and flexible way. In-line sensor technologies are required to detect the state and quality of the production in real time. To achieve a full data-driven production, each product must be tracked through the entire production as well as rapid transfer and learning of new knowledge about the products is needed.
One of the key enabler in achieving the goals in the industry 4.0 strategy is visual computing, that is a synonym of Computer Graphic and Computer vision technologies. *"Visual Computing is understood as the entire field of acquiring, analysing and synthesising visual data by means of computers which provide relevant-to-the-field tools"* [PTB+15]. The main concept of utilizing visual computing in next generation industrial production systems is that bridging computer vision with computer graphic enables us to accurate digitize the production process by utilizing modern 3D reconstruction and tracking techniques. In opposition to current industrial systems, that only make limited use of 3D computer vision technology to capture geometrical data, an introduction of in-line measurement of product geometry and/or radiometry will allow the usage of unique object models in the entire data-processing chain. By allowing industrial production systems to store knowledge about each object in every step of the production chain, we not only supply object recognition and pose estimation algorithms with prior knowledge, the industrial production systems supply all kinds of software systems, which are using geometric data with prior knowledge and status about current products. This includes robot motion planning systems, methodology systems for in-line quality control, 3D process visualization technology, Manufacturing Execution Software (MES) and many more.

Figure 1.1: Visual computing as a part of the Industry 4.0 strategy. [KWH13]

The focus in the Industry 4.0 strategy is mainly full production sites and how the individual robots and production machinery are interconnected and sharing knowledge. However, the prerequisite for sharing meaningful data, which is transfered to knowledge with the correct interpretation, is that the data is available from each robot or machine. Data which is not available in the industry today. Mainly because standardization is still needed in the field and because the individual technologies is not delivering the required data e.g. computer vision systems. In order to deliver meaning full data about objects, each robot needs to possess a capability to autonomously detect and handle the objects on the lowest level. A capability that exists today but requires configuration and programming each time new objects have to be detected. Recent research like the RoboHow project [1], KnowRob [2] [TB13], RoboEarth [3] [WBD+11] and RoboSherlock [BBBB+15] all try to find methods where robots autonomously acquire relevant object information from other sources like other robots or the internet. These research activities point in the direction of the fully autonomous perceiving robots, which will fulfill the requirement for industry 4.0. However, the research only considers object from our everyday life like cylinders and boxes with many features and not industrial objects, which can be difficult to detect. In the development towards the full autonomous production robot, something in-between is needed where the production workers are capable to instruct the robot to detect and handle novel objects. When the robot is taught how to detect and handle the object, this knowledge can be shared with other robots in the Industry 4.0 paradigm. This thesis deals with the challenges that exists in instructing and re-configuring robot vision system and how to develop better 3D sensors and object recognition algorithms.

1.1 Thesis Motivation

European manufacturing industries are challenged due to increasing demands on flexibility and changeability to maintain competitiveness. The marked for manufacturing is becoming more and more dynamic, which on the other hand requires many product changes, variety and customizations. Typical, these rapidly changing requirements have to be fulfilled without any additional product cost. This results in productions where a high mix of products with a low volume is typically. This increases the requirements for the production equipment to be reconfigurable and flexible, to cope for the many changes. These demands have increased in the last 5 years as a consequence of the financial crisis, where

[1] https://robohow.eu/
[2] http://www.knowrob.org/knowrob
[3] http://roboearth.org/

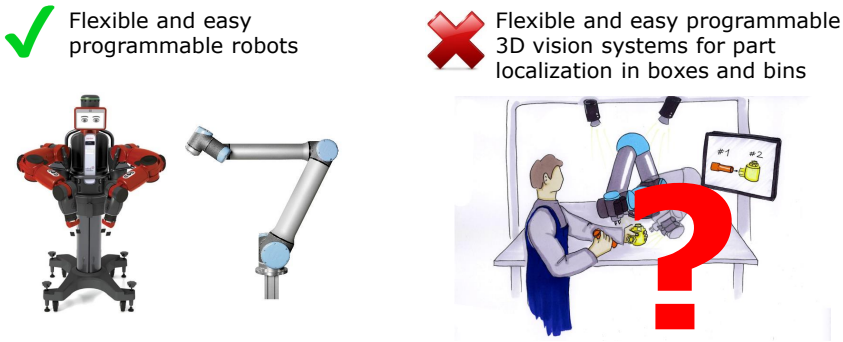


Figure 1.2: Thesis motivation.

countries like e.g. Denmark has to increase the productivity all over the society. One of the major factors for introducing flexible and agile production- and robotics systems in small and medium sized enterprises (SMEs) is the ability to (re)configure the production in order to produce new prototypes fast and easy. This, results in a shorter time to marked, ability to optimize the product design for manufacturing, which gives a more rapidly and adaptable research and development process. Furthermore, the knowledge of producing the products is kept in Denmark which reduces the risk that companies outsource the production when scaling up to full production. In addition flexible and (re)configurable robotic systems are an enabler for higher productivity in small and medium sized enterprises (SME), because of the higher throughput that a flexible production provides [GC11] [4]. This creates jobs and employment in the entire value chain. In manual labour and assembly, employees tend to fail in even simple tasks. This affects the quality of the final product and as a consequence more products with fails are lost with an economic consequence to follow. Introduction of automated production leads to fewer products with errors, which has a positive impact on the economic.

During the last five years we have seen more collaborative robots introduced on the marked. This enables companies to bypass the typical pipeline for investing in robot technology and buying their own robot without the need of professional robot integrators. With collaborative robots, companies get robots which are able to operate without fence and are easy to program. The collaborative robots give back the power to the machine operator. With collaborative robots small and medium sized enterprises (SMEs) are able to implement robots by them-

[4] https://industrialmachinerydigest.com/industrial-news/case-studies/trelleborg-selects-universal-robots/

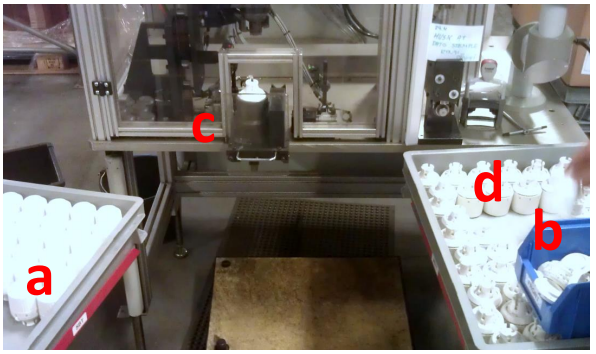
selves in their own production. This creates instant value by automating trivial tasks; but the robot is only able to move objects blindly from fixed positions. In order to enhance the capabilities of the robot, computer vision and sensor technologies are required to be able to grasp objects that are not in fixed positions. In this procedure there is no product on the marked that directly gives an end-user the possibility to easily integrate a vision system into a production without any computer vision knowledge. Smart cameras exists, which are easy to program but how to select the right camera and program the camera to reliable detect objects every time is a challenge, especially when the objects are not separated in boxes and bins. In order to build a vision system that is able to do this you need handle issues like object surface properties, light settings including controlling ambient light, calibration, selection and adjustment of camera lenses, intrinsic and extrinsic camera calibration, communication with the robot, pose transformations etc. Taking the next step towards intelligent vision systems and endow collaborative robots with sensing skills where limited knowledge from humans is needed, it is required to solve several challenges. Some of the challenges are listed below.

- Simpler integration of computer vision systems with collaborative robots
- Easy or self calibration procedures (or even no calibration)
- Easy training of new novel objects
- Visual learning of perception skills

1.1.1 Case studies from industry

Danish Technological Institute has during the last four years been in contact with many Danish and European companies that need robot solutions and help to automate part of their production. Common for many of the companies is that it is mainly final assembly where minor objects have to be located, grasped and assembled. The cases that the companies present are mostly manual labour, which they want to automate. Typically, the cases demand a high degree of flexibility, which requires that the robots are re-programmable in order to adapt different production scenarios. In this section a brief overview of some of the different cases are presented. The presented cases in this section are selected to illustrate some of the common vision related challenges that a vision system for robot guidance must handle.

The main reason why the processes are not automated yet is because no robot solution exists, that is cost efficient compared to the amount of products manufactured each week. Typically, the specific products are manufactured around



a. Pick part [a] and put it into [c]

b. Pick part [b] and put it into [c]

c. Activate machine

d. Place part at [d]

Figure 1.3: Example of simple machine tending cases. The task is fully manual today

1 to 3 days each week in average. This order based production is very common in Denmark and Europe and challenges the production equipment to be fast to (re)configure.

An example of a simple machine tending task is presented in Figure 1.3. In this task thermostats for radiators have to be assembled. The process is simple and it is easy to automate with conventional automation technology. However, it is a product that is manufactured around 2 days a week. The objects are white plastic part with a simple shape where conventional 2D machine vision could do the job in location the objects. The objects are even structured in the boxes. Nevertheless, configuring, programming and calibrating a 2D or 2.5D vision application requires vision knowledge at engineering level. The only challenging task in this process is the picking of the small plastic rings in position **b**, which requires a 3D picking solution e.g. bin-picking. However, as a first step towards an automated solution a designated dispenser mechanism could be designed for the purpose. Still, a 3D picking solution is needed in the future if we want robots to replace the human worker without a lot of expensive hardware development. A video of a similar case solved by DTI is found here[5]. Note that the work presented in the video is not a part of this Ph.D. although the applied technologies are similar.

The second case to highlight is somehow similar to the first case. Today, it is a manual machine tending case as described in Figure 1.4. The one thing to

[5] `https://www.youtube.com/watch?v=gwvfMVziEqo`

a. Pick part [a]

b. Pick part [b] and put onto [a]

c. 2 x place in machine [c]

d. Place in box [d]

Figure 1.4: Machine tending cases with geometric simple and reflective objects. The task is fully manual today

note is the objects which are reflective, shiny and geometric simple shapes. This case is a challenging task to automate because it requires not only 3D picking of parts from boxes but the object surface and geometry is very challenging for a 3D vision system. If a 2D or 2.5D single camera vision system is selected additional light sources is required in order to pick the objects reliable.

The third case is the classic bin-picking scenario where randomly positioned objects must be picked from bins or boxes. Many of the cases are machine tending tasks where objects are picked from bins and some are picking tasks for final assembly operations. In Figure 1.5 a sample of the many cases are illustrated.
Experience from many of these cases imply that a general Plug'n'play 3D vision system suited for flexible automation needs to have certain features as stated below.

- Robust towards reflective objects and changed scene illumination
- Localization of objects with few geometrical features

Figure 1.5: Examples of objects that companies want to handle with a robot. It is often objects with reflective surfaces and without texture that are placed chaotic in boxes and bins. Another characteristic is that the objects are typically simple in their geometry without many unique shape features.

1.2 Thesis Objective

The objective of this thesis is to investigate how vision systems and especially 3D vision system can become more general. To be able to solve many different tasks with a robot the visual system of the robot needs to be able to detect a larger variety of objects. Furthermore, it is important that a vision system is easy to program without the need for adjusting and tune many different parameters in order to make the detection to work. In part of this Ph.D. project the goal is to develop sub-parts of a 3D robot picking system where only a CAD model is required in order to make a robot pick objects. This is the main goal. Getting a robot vision solution where the 3D sensor acquires the relevant scene data despite surface properties and an object recognition system which only requires a CAD model and no extra parameter settings. With such a system, production staffs are able to change the system to detect new novel objects very fast.

The stated objective is a challenging task and therefore the objective is spilt into smaller parts. In this Ph.D thesis some of the most obvious challenges are identified, which include:

1. Small and robust 3D sensors able to reconstruct industrial objects
2. Re-configurable vision systems / training of 3D models
3. Robust object pose estimation of objects

1.2.1 Working thesis

Re-configurable vision systems
The most difficult task in the integration of vision systems and industrial robots is to calibrate the entire systems. In order to get all coordinate transformations between the robot and a vision system computed a calibration routine is needed. A task which is almost impossible for people without an education in robotic. Especially, if no supporting tools are available. Having a robot with a number of skills, which is able to make this calibration procedure automatically will make the integration easier. With a system that already has all transformations in place, it is a lot easier to instruct the robot to detect objects in 2D, 2.5D and 3D in single camera applications.

Working thesis 1:
"Integration of visual 2D robot guidance in a skill based framework makes instruction of the visual process easier."

Robust 3D estimation of industrial objects
Despite recent advances within 3D estimation, major challenges still exist before sensors are able to robustly and accurately estimate the 3D structure of a scene with industrial objects. The reason for this is twofold; First, the 3D estimation techniques existing today provide accurate point clouds but lack the capability to estimate surfaces with problematic surface properties e.g. specular and non-lambertian surfaces. Second, the sensors suited for use in industrial automation are still too large in size. As the demands for more flexible and agile manufacturing systems increase, the need for small lightweight sensors combined with user-friendly perception systems, increases. In particular, sensors that are directly mounted at the end-effector of a small lightweight cooperative industrial robot, are needed to make flexible and agile systems for automating tabletop assembly processes. Third, the small price friendly sensors available on the

marked today are still too inaccurate and provide noisy measurements (e.g. a Universal Robot robot manipulator).

Working thesis 2:
"Advances in state of the art structure light sensor technology, High dynamic range scanning and 3D reconstruction methodology will make it is possible to estimate specular and non-lambertian surfaces of industrial objects, which results in a dense point cloud representation of the scanned objects with sensor technology suited for small collaborative robots"

Parts of the research conducted in this industrial ph.d. project has develop a new sensor and applied novel 3D reconstruction methodologies, which increase the robustness of 3D sensors, in terms of ambient light suppression, reconstruction of specular surfaces, increasing dynamic range and robustness toward interreflections. This topic is covered in Chapter 3

3D model learning of industrial objects
Until now, existing robotic solutions in industry mostly are applying 2D image based recognition and pose estimation with a single camera mounted in the tool at the robot. However, these techniques possess some challenges. The training of the recognition algorithm and robot cell calibration can be difficult for machine operators to carry out. These increasing demands for robot vision system to be self-learning systems and easy to (re)configure, imply a demand of learning perception models. Learning the object representation directly in the production removes a long cumbersome configuration and parametrization step of a robot vision system.

Working thesis 3:
"It is possible to infer the underlying 3D structure of industrial objects by applying novel 3D sensor technology, robots and 3D registration algorithms to create full 3D models of industrial objects in a quality that makes the models usable in robot perception."

This industrial ph.d. project has investigated methods for 3D model acquisition in robot applications. The work is motivated from the question: how accurate is a 3D model required to be in order to be useful? A solutions to facilitate easy learning of object models is proposed in this thesis. This topic is covered in Chapter 4. In this chapter questions like; How do shop floor workers train new objects in an easy way in 3D? How accurate and reliable can the pose of

objects be estimated with the learned models as prior models?

3D Pose Estimating
6D Pose estimation from 3D data is still an immature research field compared to pose estimation from 2D images. Currently, the research in 3D object pose estimation is focused on designing discriminative 3D shape features that are robust towards scale and are computational unique. Many different shape features have been proposed, but many of them are not generalizing very well and only suited for specific geometries. The research community in 3D pose estimation is continuously working on handcrafting new shape descriptors. This scientific work is typically evaluated by either recording a small evaluation dataset or by using one of the existing small datasets. This results in a good evaluation of the particular feature evaluated on their own small dataset. The problem is that it does not give any answer whether the feature is generalizing better than the previous proposed features, simply because a large dataset and evaluation benchmark are missing in the research community.
In this industrial ph.d. project a new dataset and evaluation benchmark will be developed to clarify this problem and give directions on how 3D shape features should be designed to generalize well. We will record a large-scale dataset for evaluation 3D shape features and pose estimation algorithms.

Working thesis 4:
"A large-scale dataset to benchmark state-of-the-art 3D pose estimation algorithms will bring new knowledge to the research community on how 3D shape features are generalizing."

1.3 Contributions

In this chapter, a short exposition of all papers completed during this Ph.D is given. A complete list of all produced papers is found below, in Section 1.3.1. In the following sections a brief description of the individual papers are presented. Note that the descriptions are abstracts from the papers included in this thesis.

1.3.1 List of publications

During this Ph.D a total of 6 papers have been written. Five of the six papers are part of this thesis. The five of the six papers have been peer-reviewed and accepted. The last paper, [contribution F], is under review to be accepted for the *4th International Conference on 3D Vision, held from October 25th - 28th 2016 at Standford University, California, USA*. The review is a double-blind review process. Contribution B-F are included in this thesis. [Contribution A] is included in Appendix A. A complete list of all papers written, are listed below:

Contribution A:
Thomas Sølund, Rasmus Hasle Andersen, Anders Billesø Beck, and Henrik Aanæs. Combining 3D Object Modelling and Robot Skills for Intuitive Instruction of Robotic co-workers. In *2nd AAU Workshop on Robotics*, 2013. Peer-reviewed

Contribution B:
Rasmus Hasle Andersen, Thomas Sølund, and John Hallam. Definition of Hardware-Independent Robot Skills for Industrial Robotic Co-workers. In *IEEE/RSJ International Conference on Intelligent Robots and Systems 2013 - Workshop on Robotic Assistance Technologies in Industrial Settings (RATIS)*, pages 1–7, Tokyo, Japan, 2013. Peer-reviewed

Contribution C:
Rasmus Hasle Andersen, Thomas Sølund, and John Hallam. Definition and Initial Case-Based Evaluation of Hardware-Independent Robot Skills for Industrial Robotic Co-Workers. In *Proceedings of 41st International Symposium on Robotics (ISR/Robotik 2014)*, pages 101–107, 2014. Peer-reviewed

Contribution D:
Kent Hansen, Jeppe Pedersen, Thomas Sølund, Henrik Aanæs, and Dirk Kraft. A structured light scanner for hyper flexible industrial automation. In *2014 2nd International Conference on 3D Vision*, volume 1, pages 401–408, Dec 2014. Peer-reviewed

Contribution E:
Thomas Sølund, Thiusius Rajeeth Savarimuthu, Anders Glent Buch, Anders Billesø Beck, Norbert Krüger, and Henrik Aanæs. Teach it yourself - fast modeling of

industrial objects for 6d pose estimation. In *Computer Vision Systems - 10th International Conference, ICVS 2015, Copenhagen, Denmark, July 6-9, 2015, Proceedings*, pages 289–302, 2015, Peer-reviewed

Contribution F:
Thomas Sølund, Anders G. Buch, Norbert Krüger, and Henrik Aanæs. A large-scale 3d object recognition dataset. In *2016 4th International Conference on 3D Vision*, Submitted and under double-blind review. Paper notification date is the 31th of August 2016

1.3.2 Contributions in re-configurable vision systems

In **contribution B**, the framework of the DTI Robot CoWorker system is presented. The paper focuses on the challenges in robot integration that exist today and how robot programming are made easier by separating concerns. This is achieved by introducing primitives and skills, which are hierarchical composed to create a robot applications. Each skill is composed of a set of robot unit actions called primitives. These primitives include generic actions to locate object and estimate the pose of objects with one camera mounted in the robot tool. The vision primitives in the system wraps complicated pose estimation and object detection algorithm from the user. This approach makes configuration and integration of the computer vision task in an automation application easier by hiding the difficult configuration and programming from the user.

Abstract from contribution B:
"In this paper we present a framework which facilitates easy and intuitive robot instruction, allowing non-experts to instruct and use industrial robots. The framework is based on flexible, generic and hardware-independent robot Skills based on predefined symbolic unit actions called Primitives. We demonstrate the feasibility of our approach through case studies of real industry tasks which are not automated today, because they would be too expensive given the high cost of (re-)configuration using current automation approaches." From [ASH13].

In **contribution C**, the DTI Robot CoWorker is further extended and in-depth use case evaluations are implemented. **Contribution C** is a conference paper whereas **contribution B** is a workshop paper.

Abstract from contribution C:
"We propose a hierarchical action framework which facilitates easy and intuitive robot instruction, allowing non-experts to instruct and use industrial robots. The framework is based on flexible, generic and hardware-independent robot Skills, which are executed through the use of a Robot Virtual Machine. We demonstrate the feasibility of our approach through case studies of real industrial tasks which are not automated today, due to the high cost of reconfiguration." From [ASH14].

1.3.3 Contribution in 3D Estimation

In **contribution D**, a small and lightweight 3D optical sensor is presented. The sensor is a structured light sensor composed of a DLP projector and three cameras. The objective of this paper is to construct a sensor, which is small enough to be mount in the tool of a collaborative robot e.g., Universal Robot UR5 or Kuka LWR. The reconstruction algorithm combines recent advances in order to increase the robustness toward projector defocus and inter reflections. This work is motivated from experiences early in the Ph.D project where empirical results showed that the quality of point clouds from conventional 3D sensors are deficient when reflective objects are reconstructed. However, 3D sensors that are able to give reliable measurements independent of surface properties is necessary in order to pick the objects in Figure 1.5.

Abstract from contribution D:
"A current trend in industrial automation implies a need for doing automatic scene understanding, from optical 3D sensors, which in turn imposes a need for a lightweight and reliable 3D optical sensor to be mounted on a collaborative robot e.g., Universal Robot UR5 or Kuka LWR. Here, we empirically evaluate the feasibility of structured light scanners for this purpose, by presenting a system optimized for this task. The system incorporates several recent advances in structured light scanning, such as Large-Gap Gray encoding for dealing with defocusing, automatic creation of illumination masks for noise removal, as well as employing a multi exposure approach dealing with different surface reflectance properties. In addition to this, we investigate expanding the traditional structured light setup to using three cameras, instead of one or two. Also, a novel method for fusing multiple exposures and camera pairs is given. We present an in-depth evaluation, that lead us to conclude, that this setup performs well on tasks relevant for an industrial environment, where many metallic and other surfaces with difficult reflectance properties are in abundance. We demonstrate, that the added components contribute to the robustness of the system. Hereby, we demonstrate that structured light scanning is a technology well suited for hy-

per flexible industrial automation, by proposing an appropriate system." From [HPS+14].

1.3.4 Contributions in 3D Pose estimation

In **contribution E**, a vision system that allows fast learning of 3D object models in a production scenario is presented. The objective of this work is to demonstrate and verify that visual learning of perceptual models are a valid approach in order to increase the flexibility of a robot vision system.

Abstract from contribution E:
"In this paper, we present a vision system that allows a human to create new 3D models of novel industrial parts by placing the part in two different positions in the scene. The two shot modeling framework generates models with a precision that allows the model to be used for 6D pose estimation without loss in pose accuracy. We quantitatively show that our modeling framework reconstructs noisy but adequate object models with a mean RMS error at 2.7mm, a mean standard deviation at 0.025mm and a completeness of 70.3% over all 14 reconstructed models, compared to the ground truth CAD models. In addition, the models are applied in a pose estimation application, evaluated with 37 different scenes with 61 unique object poses. The pose estimation results show a mean translation error on 4.97mm and a mean rotation error on 3.38 degrees." From [SSB+ed].

In **contribution F**, a new large scale object recognition dataset is presented. The objective of this work is to provide the 3D object recognition research community a new dataset for evaluation of local shape features and 3D pose estimation algorithms. During this Ph.d. it has become clear that fundamental challenges exist in 3d object recognition and pose estimation of geometric simple shapes, which often have no or limited texture e.g. cylinders and flat objects. However, two comparison studies form 2016 [GBS+16],[BPK16] show very high matching performance with state-of-the-art local shape features. Results that are achieved because the datasets used is to small in term of the number of object and included in the scenes. Furthermore, the included objects are ideal and with a lot of shape features. In this contribution a more realistic dataset that represents the real world problems in robotic and industrial automation is proposed and an evaluation of existing local shape features are conducted. It is the hope that the proposed dataset will push state of the art towards algorithms and methods that enables detection of objects with dissimilar surfaces. A property that is essential if plug-n-play 3D vision systems for robot picking

should be realised.

Abstract from contribution F:
"This paper presents a new large scale dataset targeting evaluation of local shape descriptors and 3d object recognition algorithms. The dataset consists of point clouds and triangulated meshes from 292 physical scenes taken from 11 different views; a total of approximately 3204 views. Each of the physical scenes contain 10 occluded objects resulting in a dataset with 32040 unique object poses and 45 different object models. The 45 object models are full 360 degree models, which are scanned with a high precision structured light scanner and a turntable. All the included objects belong to different geometric groups; concave, convex, cylindrical and flat 3D object models. The object models have varying amount of local geometric features to challenge existing local shape feature descriptors in terms of descriptiveness and robustness. The dataset is validated in a benchmark, which evaluates the matching performance of 7 different state-of-the-art local shape descriptors. Further, we validate the dataset in a 3D object recognition pipeline. Our benchmark shows as expected that local shape feature descriptors without any global point relation across the surface have a poor matching performance with flat and cylindrical objects. It is our objective that this dataset contributes to the future development of next generation of 3D object recognition algorithms. The dataset will be made public available together with this paper." From [SBKA16].

1.4 Thesis structure

This thesis is organized as an anthology, where each captures gives a general introduction and a deeper technical description of the theory of the problem domain. Each chapter ends with the relevant papers. The rest of the thesis is organized as follows; In the following three chapters the related work for each problem domain is presented individually. Chapter 2 deals with the work on visual robotic guidance of re-configurable robotic system. In Chapter 3 the work regarding 3D estimation is presented. Whereas the work on model learning and pose estimation are presented in Chapter 4. In Chapter 5 a discussion of future work, perspectives and a conclusion is presented.

Chapter 2

Flexible Automation

2.1 Introduction

In this chapter, techniques for detecting and computing the pose of rigid objects in a robotic work cell are presented and discussed. The focus will be methods for detecting objects with computer vision algorithm working with 2D image data. This is still the dominating computer vision technology for robot guidance in the industry today. In this chapter the individual steps of a pose estimation pipeline based on local image features will be covered; from feature detection and description in Section 2.4 to matching and estimation in Section 2.5 and 2.6. A brief overview of existing commercial machine vision technologies is presented in Section 2.2. A critical problem in every robot guidance application is the extrinsic calibration of the work cell such that a transformation of poses computed in the local camera reference frame easily is transformed to the robot reference frame. This topic is covered in Section 2.3. In Section 2.7 we present state of the art object detection systems for robots that implement the pipeline. In the following Section 2.8, a detailed description of the vision system implemented into the DTI Robot Co-worker which enables fast reconfiguration of simple automation tasks is given. This work is published in [Contribution B] and [Contribution C] which is enclosed in Section 2.9. The chapter is completed with a conclusion in Section 2.10.

In general, object recognition is the task of recognizing whether a particular object is present in an image and pose estimation is the task to precisely locate the object with a position and a rotation. Methods for detecting objects are divided into local and global methods. Global methods using low-level image descriptors based on the appearance like color or texture histograms and models. Global appearance based methods are mostly used in object recognition task where the presence of an object in an image has to be decided. Global appearance methods typically needs a good segmentaion of the whole object which limits the performace in presence of clutter, occlusion or background changes. This limitation makes global methods less applicable in industrial robotics where objects typical are occluded in clutteded scenes. Local methods detect interest points and compute local feature descriptors by considering the local pixel values around the feature. These features are then matched with a known model of the object that has to be recognized and/or detected. Models are typical 3D CAD models, trained template or feature models which are trained from example images of the object of interest. The focus in this chapter is techniques for stationary object recognition and not temporal estimation techniques where objects are recognized in e.g. a video sequence. Methods and proposed work for object classification and object class recognition based on local image features are not considered in this thesis. For detailed information about object class recognition, see recent surveys by Zhang *et al.* [ZYH+13]. Moreover, the review will not include methods based on global appearance like subspace methods but focus exclusively on feature-based methods for object detection and pose estimation. More information about global appearance method is found in [RW08].

In this chapter single camera object recognition using geometric edge relations as model e.g. matching of edges from a CAD model is not considered. 3D to 2D projection like CAD matching is in general used in industrial robotics as a good method for 3D picking of object where significant edges in the image are present. However, the focus of this thesis is mainly feature based methods applied on image and point cloud data and robust 3D sensors. More information regarding the early work in this field is given in [Low87], [Mun06] and recent work i.a. [UWS09],[CC12].

2.2 Commercial machine vision - a review

Imaging methodologies in computer vision applications in robotic are typical split into two categories; 2D and 3D imaging. 2D imaging utilizes a single camera to create an image in either gray scale or color, as commonly known from

consumer cameras and video camcorders. Applications for 2D Machine vision varying from online Character Recognition (OCR), 1D/2D code reading, checking of labels and package, quality inspection, meteorology to Robot Guidance, see Figure 2.1. 3D vision technologies imaging the world with many different technologies but in the end the 3D sensor creates a 3D image that not only includes a pixel (x,y,intensity) but a depth of each pixel. This review will focus on existing commercial 2D machine vision solution for robot guidance. Later in Chapter 3.2 and 4.2, a brief overview of existing commercial 3D systems are presented.

Today, we can divide commercial 2D vision solutions that utilize a single cam-

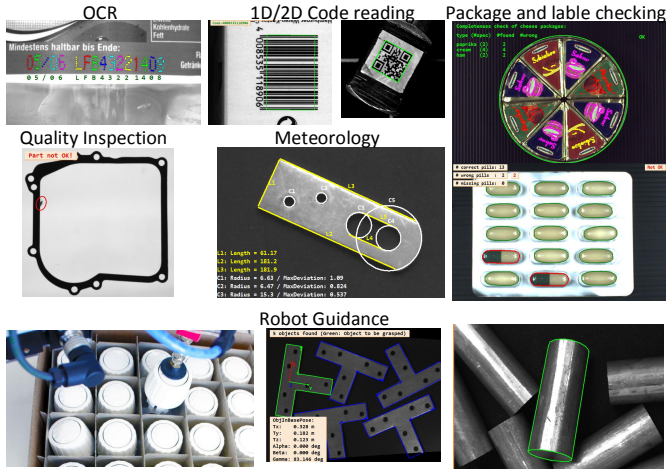


Figure 2.1: Industrial machine vision covers the need in many application. Online Character Recognition (OCR), 1D/2D Code Reading, Checking of labels and package, Quality Inspection, Metrology and Robot Guidance are the most common uses.

era in the manufacturing industry into three different categories; vision sensors, smart cameras and vision systems, see Figure 2.2. Vision sensors are imaging devices with dedicated purposes such as online character recognition (OCR) or code reading. These types of sensors are inexpensive, easy to configure and deploy for machine builders. Integration of the device in a production is more or less Plug-n-Play. Examples of these kind of sensors are SICK Lector, Omron and Banner P4. If vision sensors do not have the required functionality for the application, the smart cameras provide more flexibility. Smart cameras are script-able cameras that allow an application engineer to customize a vision algorithm for a special purpose by combining predefined functions such as finding shape models (2D pattern recognition), blobs and different methodology tool etc. Smart cameras are often general purpose devices with limited functionality.

Some recently introduced smart cameras includes all the complex functionality of comprehensive machine vision library but in a practical housing which is easy to install. However, these new smart cameras still requires technical skills to use. Often accessories like Human Machine Interfaces or webserver SDKs are available that allow easy development of simple displays to show the vision results. It requires some knowledge from the machine builder to integrate smart cameras in a robot application. Common for both vision sensors and smart cameras is that the data processing is running onboard, with no need for external computing devices. This enables fast integration with a PLC or Robot. Many of the manufactures of smart cameras delivers functional blocks for different PLCs e.g. Siemens or Beckhoff which make the integration of the camera even simpler. Issues like reliable detection under different lightning condition, calibration and coordinate transformations are challenges that still exist with these systems and have to be handled by technical skilled persons. When vision sensors and

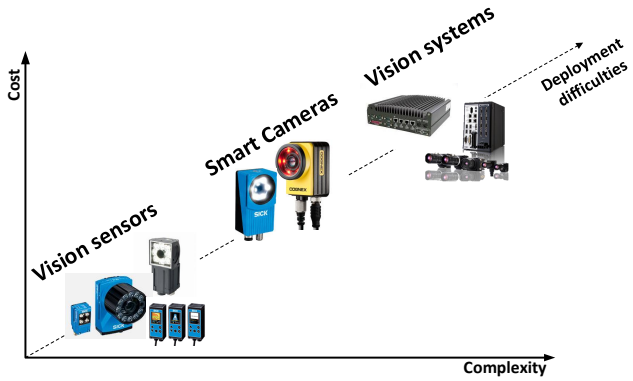


Figure 2.2: Different categories of 2D vision systems and their complexity vs. cost.

smart cameras are not providing the required flexibility and functionality for a given application, vision systems are the only alternative. Vision systems are systems that include industrial cameras, a computational unit like an industrial PC, machine vision lightning and one of the comprehensive professional image processing libraries like Halcon[1], Cognex [2], Matrox[3], Scorpion[4] etc. With a vision system, a vision engineer is able to customize a vision solution for a given

[1] http://www.halcon.com/
[2] http://www.cognex.com/
[3] http://www.matrox.com/imaging/en/
[4] http://www.scorpionvision.com/

application. The development of a vision system for a factory automation application requires deep knowledge within programming, computer vision, cameras and lightning technology but provides the full flexibility.

2.3 Robot guidance

In machine vision, three different computer vision approaches for computing pose of an object in a work cell exist; 2D, 2.5D and 3D pose estimation. Standard 2D vision applications are able to measure and detect objects in one plane (x,y,R_z) and no height information is measured. Most of the current vision systems for robot guidance fall into this category. This could be application where objects are picked from e.g. a table, conveyor or at the bottom of a box, where the distance from the object to the camera is constant. The nature of 2D vision applications requires that the third dimension is inferred from the 2D image. 3D pose coordinates are computed by extrinsic camera calibration such that the world coordinate frame is known. The world coordinate is typically called the object frame in order to have a frame in the robotic system that represents the object. The object frame is placed in the center of the camera calibration target e.g. a chessboard calibration target, see Figure 2.4 (green frame). Only one feature or contour pattern is needed in order to solve the perspective transformation problem in Equation 2.1 that computes the 3D pose of the object in relation to a known object frame and camera intrinsics.

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} R & t \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2.1}$$

where λ is the scale factor, (u,v) are the image coordinates of the feature or centroid of a trained pattern, K is the intrinsic camera matrix from a separate camera calibration procedure, $[R,t]$ is the camera reference pose measured e.g. with a calibration target, z is the known elevation of the feature point in relation to the plane $z = 0$.

The general requirements for 2D robot picking applications are that the objects have to be flat in order to be accurate, because the depth of the (x,y) point is determined by the calibration target. Alternative, the calibration target much be placed on top of the object or in the depth where the robot should grasp in order to ensure the precision of the pose. Another limitation of 2D applications

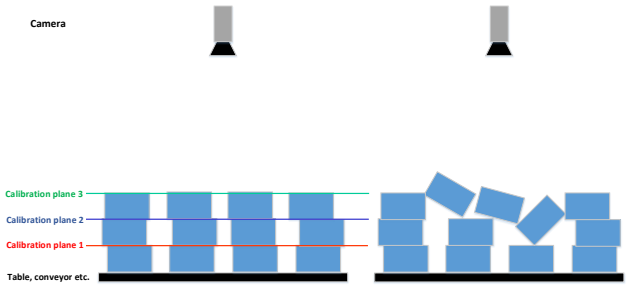


Figure 2.3: 2D Robot guidance operating in one calibration plane. **Left:** With separated layers of objects it is possible to calibrate the height for each layer such that 2D picking is possible. **Right:** If the object are tilting it is not possible to detect and compute a valid pose based on the calibration planes.

is that picking from different layers it is associated with several calibration procedures because the depth is needed for each layer, see Figure 2.3 left. Second, objects that are in different layers could tip over such that they cannot be detected because of perspective distortion, Figure 2.3 right.

In robot picking applications where the (x,y) coordinate and a rotation R_z is not enough but the height z is required, 2.5D robot guidance is the solution. Applications where 2.5D robot guidance is required include solutions where objects are picked from a table, conveyor or from boxes or pallets, where the distance from the object to the camera is not constant. 2.5D robotic guidance is well suited for applications where objects are layered e.g. in a pallet. In order to compute the extra dimension two feature points (u_1, v_1), (u_2, v_2) must be detected and its 3D coordinates P_{w1}, P_{w2} with respect to the object coordinate system have to be known. With an intrinsic calibrated camera, the 2.5d pose (x, y, z, R_z) are computed by solving the linear system of equations in Equation 2.2.

$$\begin{bmatrix} & -u_1 & 0 \\ K & -v_1 & 0 \\ & -1 & 0 \\ & 0 & -u_2 \\ K & 0 & -v_2 \\ & 0 & -1 \end{bmatrix} \begin{bmatrix} T_{object} \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} -K & R_z & P_{w1} \\ -K & R_z & P_{w2} \end{bmatrix} \tag{2.2}$$

where (u_1, v_1), (u_2, v_2) are the detected image coordinates, K is the intrinsic camera matrix, T_{object} is the object pose (x, y, z) which we want to estimate,

λ_1, λ_2 is a scale factor, R_z is the camera reference rotation around z-axis and P_{w1}, P_{w2} are the known 3D world points of the two feature points.

For applications where the full 6D pose of objects is needed, at least 4 feature points must be detected in the image and the correspondent four world points must be given. The object pose is estimated with a Perspective-n-Point algorithm that solves perspective transformation in Equation 2.1 with unknown $[R|t]$. The Perspective-n-Point problem will be further discussed in Section 2.6.2. In general full 3D pose estimation is typically not available in smart cameras but only through advanced image processing libraries like Cognex, Halcon and OpenCV that is implemented in a vision system.

Commercial systems that provide functionalities for 2D robot guidance include both smart cameras and vision systems. SICK [5], Cognex [6], Teledyne Dalsa [7], ADLink [8] and Microscan [9] are some of the major brands providing smart cameras for the industry. Some of the smart cameras e.g. SICK PIM 60 allow you to easily align the camera measurements to an external coordinate system e.g. the robot base frame, in the same way as described above where 3 points are touched with a calibration tool.

2.3.1 Alignment

Computing the pose in the robot coordinate system e.g. the robot base frame or robot tool frame, requires a known transformation between the robot base frame and the object frame. The robot frame to object frame calibration is performed with a calibrated tool mounted in the tool of the robot e.g. a tip. The tool center point (TCP) of the calibration tool can either be placed at the tip or in the robot flange. Now it is trivial to get the robot base to object frame transformation (green to red frame in Figure 2.4) by touching 3 points on the camera calibration target that correspond to the origin, a point on each x and y axis of the target (green frame in Figure 2.4). Many robot controllers offer this as a build in function, which allows you to create user frame e.g. in ABB robots this procedure is named work object calibration. The transform between object and robot tool frame is found by solving an absolute orientation problem that recovers the rigid body transformation between the two coordinate systems.

Often 2D and 2.5D robot guidance applications lack the required precision due to robot inaccuracy and bad calibration of the lens distortion. A common way

[5] https://www.sick.com
[6] http://www.cognex.com/
[7] https://www.teledynedalsa.com
[8] http://www.adlinktech.com
[9] http://www.microscan.com

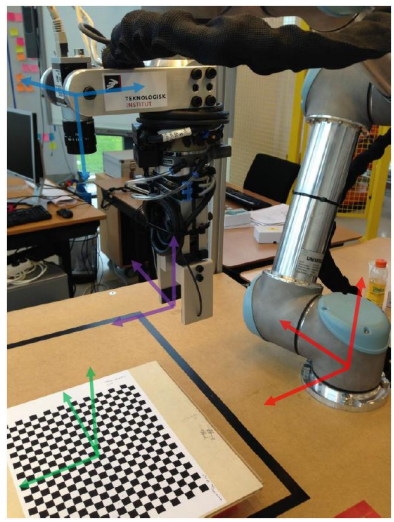


Figure 2.4: Required calibrated frames for single camera pose estimation. Green is the object frame, red is the robot base frame, blue is the camera frame and purple is the Tool Center Point (TCP) frame.

to handle these problems are to create a close loop alignment procedure, where the robot iteratively tries to center the object of interest in the middle of the image. When we know that the object is in the center of the image and in one plane with a rotation around the object z axis, we can apply the same grasping transform each time. This method is widely used in commercial robot guidance applications due to the robustness. A drawback is that the robot needs some movement iterations before the object is aligned in the center of the image. In open-loop alignment the pose is computed directly and grasped by the robot.

In open-loop alignment it is required to know the transformation from camera reference frame to the robot tool frame. This transformation is computed in a hand-to-eye calibration procedure. The result of the hand-eye calibration is used to transform a computed 3D pose in the camera reference frame into robot base frame or other robot reference frame. This transformation generates the appropriate robot pose that is send to the robot for grasping an object. The camera is mounted either as a tool mounted camera or a scene mounted camera where the camera is static mounted above e.g. a table or conveyor. For the stationary camera configuration a 3D pose in the robot base frame is computed as $T^{Robot}_{Object} = T^{Robot}_{Camera} \cdot T^{Camera}_{Object}$ and for the camera in tool configuration it is computed as $T^{Robot}_{Object} = T^{Robot}_{Tool} \cdot T^{Tool}_{Camera} \cdot T^{Camera}_{Object}$. Conceptual hand-eye calibration requires a list of robot poses and a list of camera poses taken from a number of

different views. In case of a stationary camera configuration, a calibration target is mounted in the robot tool and the robot moves the target to different position in the robot work cell while robot and camera poses are recorded, see Figure 2.5 (right). Camera poses are easily estimated with e.g a Perspective-n-Point algorithm, see Section 2.6.2. In case of tool mounted camera, the calibration target is placed at a stationary position in the robot work cell while the robot moves to different positions while observing the target and recording robot and camera poses, see Figure 2.5 (left).

Fundamental two different methods for solving the hand-eye calibration prob-

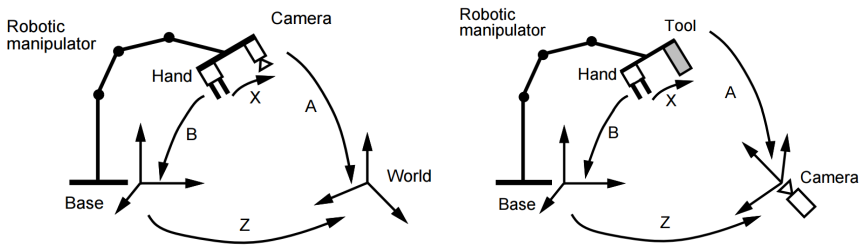


Figure 2.5: Left: Camera in tool configuration. **Right:** Stationary camera configuration [DH98]

lem exist. One that simultaneous estimates the hand-eye transformation and the pose the world reference frame e.g. the calibration reference frame by solving Equation 2.3.

$$AX = ZB \tag{2.3}$$

where A is the robot TCP pose, B is the induced camera poses, Z is the world to robot base transformation (e.g. calibration target frame to robot base frame) and X is the hand-eye transformation. Published methods that solves $AX = ZB$ include [Wan92], [ZRS94], [RDLD97] and [DH98], [SH06] and [HHP16]. Throughout this thesis we apply the method by Dornaika and Horaud [DH98].

A simpler solution to the Hand eye calibration problem is to only solve for the hand eye transformation. Thus, Equation 2.4 have to be solved.

$$AX = XB \tag{2.4}$$

where A is the robot TCP pose, B is the induced camera poses and X is the hand-eye transformation. Several established closed form solutions for the $AX = XB$ problem have been published including [TL89], [SA89], [Wan92], [PM94], [Dan98] and solutions applying non-linear optimization techniques [PM94], [HD95],

[ZS93].
The problem of hand-eye calibration is not in the scope of this thesis, altrough the algorithms are used in the scentific work during the project. If the reader needs an in-depth review of the individual methods, newer publications on this topic e.g. [SH06] and [SEH12] are recommended.

2.4 Local image features

Detection of humans and objects from images is one of the major tasks in computer vision. This research field has been going on the last 50 years and is still a major topic [AT13]. The impact of local image features during the last 30 years has been major and is now considered as the standard method for object recognition in images. The proved invariance towards rotation and scale is one of the reasons for its popularity. The robustness towards clutter and occlusion are another good property. In this section we will go through some of the basic methods for detecting and describing local image features.

2.4.1 Local Features: Detection and Description

Local invariant features detectors and descriptors are two of the key technologies in feature based computer vision research, which enables efficient object recognition and categorization approaches that are stable under different viewing conditions like view point, different illumination and partial occlusion. Local invariant features enable computer vision algorithms to reliable find local structures in an images and encode them such that they are invariant to translation, rotation, scale and affine deformation. The goal of invariant and distinct feature representation is to be able to match features between an image of an object (*the model*) and a test image (*the scene*), by having a sparse set of local images patches that capture the important and interesting structure of a image. This process in computing local invariant features are split into two steps; detection of interest points and description of the point by considering a local patch around the point. In the following we will briefly review work in computing interest points, followed by a review of advances in feature description. Recent complete reviews of interest point detection and feature description are given in [LWTD15], [MJWW15].

2.4.1.1 Interest point detection

The simplest interest point detectors are convolution based methods like the *Hessian* [Bea78] and *Harris* [HS88] detector that both capture corners like structures. The *Hessian* detector locates image points, which exhibits strong derivatives in two orthogonal directions by computing the second derivatives and search for points where the determinate of the Hessian is maximal. The *Hessian* detector finds regions with strong texture variations and corner structures. The *Harris* finds corners by locating image points where the second-moment matrix has two large eigenvalues, that corresponds to two dominant orientations. If the one of the eigenvalues is significant larger that the other, the capture structure will be an edge. The two basic detectors are remarkably robust to noise, changes in illumination and image rotation [SMB00], but sensitive towards scale. An early comprehensive review and comparition of interest point detectors are found in [SMB00]. One of the application areas where the simple yet powerful *Harris* detector is applied is in feature tracking algorithms e.g. the most known approch, the Kanade–Lucas–Tomasi KLT tracker [TK91]. Shi and Tomasi [ST94] extended the *Harris* detector by changing the scoring function that determines if a point is interesting, by minimizing the eigenvalues instead of evaluating the autocorrelation matrix. In order to be robust towards acale and affine transformation, Mikolajczyk and Schmid [MS04] proposed the *Harris-Laplace* detector based on a Laplacian of Gaussian filter and the *Harris-Affine* detector, which solves the problem of automatic scale selection and affine invariance, respectively. The idea of automatic scale selection for interest points was initial proposed by Lindberg [Lin98]. Similar to the *Harris* detector the *Hessian* detector becomes invariant to scale by finding the local maxima of the Laplacian-of-Gaussian filter and invariant to affine transformation, the *Hessian-Laplace*[MS05] and the *Hessian-Affine* [MTS$^+$05]. Both the *Harris-Affine* and *Hessian-Affine* estimates affine shape of the point neighbourhood by evaluating the eigenvalues of the second moment matrix and finds interest points with blob and ridges like structures. The *Harris-Affine* and *Hessian-Affine* detectors were the first detectors in the family of region based detectors. Other region based detectors include Intensity based regions (IBR) and Edge based regions (EBR) [TVG04], *Maximally Stable External Regions (MSER)* [MCUP02] and *Salient Regions* [KZB04]. IBR search for local intensity extrema and detect the boundary of the region by tracing lines from the point. A function of intensity differences along the lines are evaluated to find the boundary that corresponds to the extrema of the intensity difference. MSER finds the region boundary by intensity thresholding. EBR finds regions by detecting Harris corners and Canny edges. From the Harris corner, edges are followed and the region are determined by evaluating functions of intensity moments. A extensive comparison of Affine region detectors is given by Mikolajczyk *et al.* [MTS$^+$05]. Compared to previous presented corner detectors which relie on local gradients, the SU-

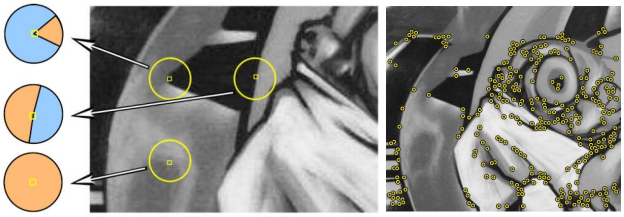


Figure 2.6: The SUSAN corner detector. From [TM08]

SAN detector proposed by Smith and Brady [SB97], segments a circular region around a point in "similar" and "dissimilar" regions based on the image intensity of the center point, the nucleus. A corner is detected by examine the ratio or "similar"/"dissimilar" area of the circular region and the centroid. Moreover, the SUSAN detector is able to detect edges by increasing the threshold for "similar"/"dissimilar" area of the circular region. If the ratio is around 50% a nucleus is an edge point and a corner point if it is below 25%. One of the downsides with the SUSAN detector is that is falls short when rotation and scaling are involved. Rosten and Drummond [RD06] proposed the Features from

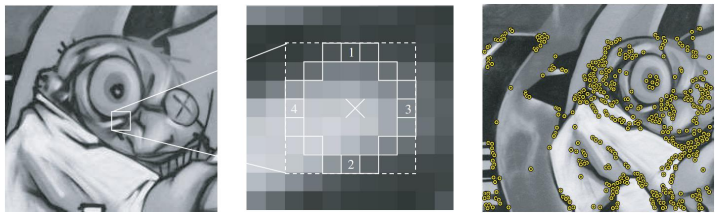


Figure 2.7: The FAST corner detector. From [TM08]

Accelerated Segment Test (FAST) detector that is an extension of the SUSAN dector. The FAST detector is computational efficient compared to e.g. SIFT and SURF. This makes it suitable for real-time applications such as Simultaneous Location And Mapping (SLAM). Klein and Murray, 2007 [KM07] use the FAST corner detector in their significant work on parallel tracking and mapping for augmented reality as well as Klein *et al.* [HKH+12] applied FAST for computing the camera motion in their popular RGBD-Mapping project, which is mapping indoor locations with a single Kinect sensor. The FAST detector runs a high-speed test of 4 pixels in a circle around the nucleus, by first comparing pixel 1 and 2 in Figure 2.7 and the 3 and 4. If three of the four points are either darker or brighter than the nucleus $\pm$ a threshold, the nucleus is a corner and then the basic FAST algorithm examine the rest of the 12 pixels. The basic

algorithm is not fast, thus a machine learning approach is applied to train a decision tree classifier, which learn the distribution of the corner configuration using the ID3 algorithm [Qui86], from a set of training images of the specific environment. The problem of multiple adjacent corner detections are handled by non-maximum suppression. Despite the popularity of the FAST detector, it not invariant to camera rotations and scale [MHB+10]. Furthermore, FAST needs to be retrained if the environment/application is changing significant. To overcome some of these drawbacks Mair *et al.* 2010 [MHB+10] proposed the "Adaptive and Generic Accelerated Segment Test" (AGAST) corner detector by changing the decision tree from a static to a dynamic adapting tree for classifying corner points. The decision tree dynamically adapts to the environment while processing image by exploring a more detailed configuration space for the subspace division (darker, not darker, similar, not brighter, brighter) and applying a customized backward induction method. Moreover, the author of AGAST showed a 50% speed up with respect to the FAST detector. In order to make the AGAST detector scale invariant Leutenegger *et al.* [LCS11] introduced in 2011 the "Binary Robust Invariant Scalable Keypoints" (BRISK), which added a scale space to the AGAST detector. The BRISK detector finds interest points by non-maxima suppression and interpolation across all scales, using the FAST score as a measure for saliency. The concept of scale space detectors will be introduced below together with the SIFT detector.

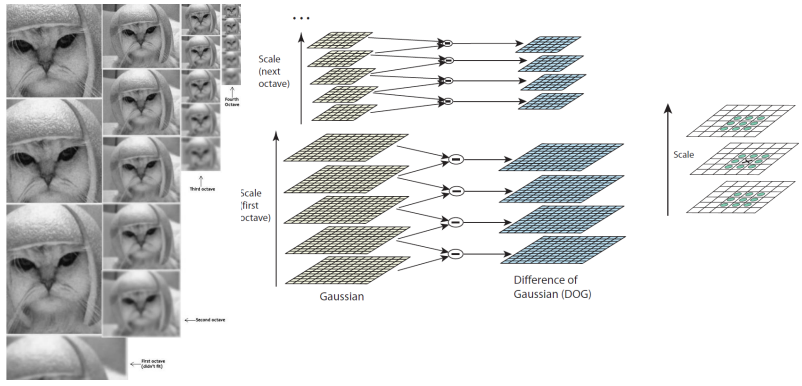


Figure 2.8: The SIFT interest point detector. From [Low04]

2.4.1.2 Scale-space features detectors

The most famous and widely used feature in computer vision is the Scale Invariant Feature Transform (SIFT) proposed by Lowe [Low04]. The SIFT feature is both an interest point detector and a feature descriptor. It was the first interest point detector that applied the Scale Space filtering method proposed by Andrew Witkin [Wit84], to detect interest points at different scales. SIFT computes the scale space by applying Gaussian blur and resize the image in five scales, see Figure 2.8 left, and compute the Difference of Gaussian (DoG) by subtracting two images, see Figure 2.8 middle, which is an approximation of Laplacian of Gaussian (LoG), but more computational efficient. Interest points are detected by checking for maxima and minima at neighbouring pixels in the current Difference of Gaussian image and at the scale above and below it. In totally 26 pixels, see Figure 2.8 right. Sub pixel accuracy is assured by the Taylor series expansion of the scale space. To get rid of low contrast and edge points, the SIFT detector compares the intensity if the maxima/minima point with a threshold and apply a Harris like corner detection by evaluating the ratio of the eigenvalues of the second moment matrix. The SIFT detector assign the gradient orientation and magnitude to each detected interest point to insure rotation invariance. In 2006, Bay *et al.* [BTVG06] proposed the Speeded Up Robust Feature (SURF), as an efficient alternative to SIFT. SURF is both an interest point detector and descriptor like SIFT. Similar to SIFT, the SURF detector uses scale space filtering and an approximation of the Hessian-matrix by applying box filters to detect interest points at image points where the determinant of the Hessian-matrix has its maxima. The box filters can be convoluted with an image using integral images with very low computational cost. Moreover, the SURF detector scales up the box filters instead of downscaling the original images. This allows for parallel compting of the interest points at the different scales by convolute the input image with the box filters at different scales. As opposite to SIFT, the SURF detector finds blobs and not interest points. The DART interest point detector proposed by Marimon *et al.* [MBAG10], further increased the speed of scale-space methods, by approximating the Hessian by piece-wise triangle filters. DART speeds-up the interest point detection and description with a factor 3 and 6 compared to SURF and SIFT, respectively. The DART feature uses the DAISY feature [TLF10] descriptor for each detected interest point. Agrawal *et al.* 2008, [AKB08] proposed the "Center Surround Extremas" feature,(CenSurE), which approximates the Laplacian of Gaussian like SIFT but are using bi-level octogons and boxes that computed with integral images, as the SURF detector. The Octogonal filters is in nature rotation invariant, which makes the method rotation invariant. The computational performance is ensured by up-scaling the bi-level filters instead of down sample image. The same approach as for SURF. This results in a feature that is more distinctive, stable and repeatable in changes of viewpoint compared to SIFT

and SURF. The CenSurE is implemented in OpenCV as the STAR feature with a minor modification to the bi-level filter. Ebrahimi and Mayol-Cuevas (2009) [EMC09] optimized CenSurE by proposing the "Speeded Up Surround Extremas" (SUSurE) by skipping the computation of the filter response if the response for the previous pixel was very low. This resulted in a significant speed-up with only minor loss of repeatability.

2.4.1.3 Local invariant feature descriptors

Interest points are able to detect points in images that exhibit interesting and meaningfull structures in an image like corners, blobs or edges. In order to match these strong local features across different images, a signature description of the points needs to be extracted; a feature descriptor. A description is typical constructed by considering the area around the point with a certain radius; called the support radius. The ideal local feature descriptors should be distinctive such that features are able to deal with a large number of objects and are robust to occlusion and background clutter. Moreover, local feature descriptors need to be invariant to both geometric and photometric transformations e.g. affine transformation and intensity change due to illumination changes. Many of the previous interest point detectors are original proposed together with a descriptor. This is the case for SIFT [Low04], SURF [BTVG06], BRISK [LCS11], ORB [RRKB11] and DART [MBAG10].

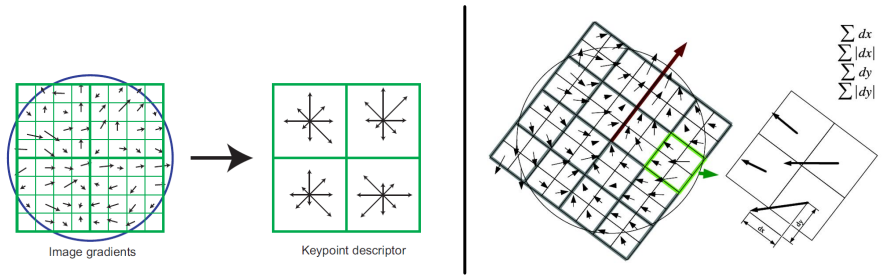


Figure 2.9: Left: Computation of the SIFT descriptor, from [Low04] and **Right:** the SURF descriptor, from [BTVG06]

SIFT is the most cited and well-known feature descriptor prooposed by David Lowe [Low04]. The SIFT descriptor computes the gradient magnitude and orientation in a 16x16 region around the interest point. The region is then divided

into 16 sub-blocks of 4x4 size. In Figure 2.9, the 16x16 region and 4x4 region are illustrated but with the half of the dimension. For each of the sub-blocks an orientation histogram with the bin size of 8 is created. In total a SIFT feature has a bin size of 128 in total. The orientation of the 16x16 region for the SIFT feature extraction is determined by the interest point orientation estimated in the detection process, which makes the feature descriptor rotation invariant. Finally, the feature vector is normalized to unit length to make the SIFT descriptor invariant to illumination changes. During the years, SIFT has proved to be very stable in many application but the relative large feature vector of 128 is a problem when applications require very fast feature matching with many features. In order to speed up feature matching Ka and Sukthanka [KS04] proposed the 36-dimensional descriptor called PCA-SIFT, which reduces the dimensionality of gradient patch in 41x41 region by principal component analysis. It has later been shown in [JG09] and [MS05], that this dimensionality reduction makes the feature crease less distinctive compared to SIFT. Despite, PCA-SIFT speeds up the matching process the feature computation is slow when applying PCA. Several extensions and new local features that are inspired by the SIFT feature have been proposed because of SIFTs proved robustness towards scale and rotation. The Gradient Location and Orientation Histogram (GLOH) [MS05] extended SIFT by computing the descriptor in a log-polar grid with 3 circular bins divided into 8 angular directions, which results in 8 + 8 + 1 bin equal 17 location bins. The log-polar grid is quantized into 4 x 4 grid such that the full descriptor has 272 bins in total, which is reduced with PCA to 128 dimensions. The 128 dimensional feature vector is computed by taking the 128 largest eigenvectors of the covariance matrix obtained from 47,000 image patches. It has been reported that the GLOH feature preforms slightly better than SIFT [MS05]. The Rotation Invariant Feature Transform (RIFT) proposed by Lazebnik *et al.* [LSP05] uses detected regions instead of corners as the SIFT feature. The RIFT feature creates a descriptor of a sparse set of detected regions that are normalized to a unit circle to reduce affine ambiguity. This unit circle is divided into four concentric rings and a gradient orientation histogram with 8 orientations are computed, which give a 4 * 8 dimensional descriptor.

The 128 dimensions of the SIFT and GLOH features make matching in object recognition tasks with a large object database time consuming. In order to solve this, Bay *et al.* [BTVG06] introduced the Speeded-Up-Robust-Feature (SURF) with only 64 dimensions. The SURF descriptors are computed in a 20sx20s grid aligned with the orientation of the detected interest point, where s is the size of the window. This region is divided in a 4x4 region as illustrated in Figure 2.9 (right). For each sub-region, the response from a horizontal and vertical Haar wavelet is computed and the SURF feature vector is constructed. Note that the Haar wavelet responses are computed efficient with integral images. If the size of the feature size is increased better distinctiveness is provided as the case with many other local features. Together with SURF, Bay *et al.* [BTVG06] proposed

the Upright-SURF which is faster to compute but not rotation invariant more than +/−15%. U-SURF is suited for applications where the camera remains horizontal. Based on the success of the SURF feature some improvements have been proposed. Agrawal *et al.* [AKB08] introduced together with thw CenSurE interest point detector, the Modified-SURF (M-SURF). "*The M-SURF is a variant of the original SURF descriptor, but handles better descriptor boundary effects and uses a more robust and intelligent two-stage Gaussian weighting scheme.*", [ABD12].

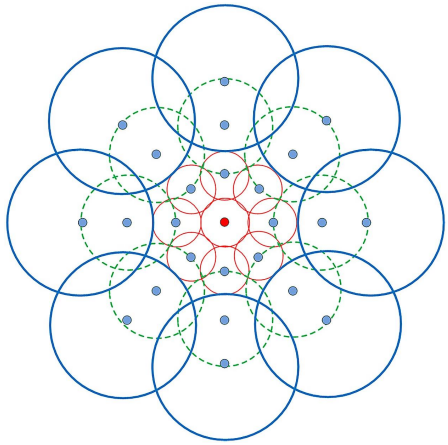


Figure 2.10: Directional diagrams of the DAISY descriptor [TLF10]

The DAISY descriptor [TLF10] was original designed for efficient dense stereo matching for wide-baseline stereo and is inspired from SIFT and GLOH. One of the design goals was to make it very fast to compute densely. In contrast to SIFT and SURF, the DAISY descriptor uses a circular neighbourhood around an interest point. This neighbourhood is divided in to 8 direction gradients. Each of the direction gradient is sampled into 3 layers of central-symmetrical circles. A total of 24 circular diagrams as shown in Figure 2.10. Each of the circular diagrams is convolved with several Gaussian kernels with 3 different standard deviations where the amount of Gaussian smoothing is proportional to the radii of the circles. The DAISY descriptor is constructed from the 8 different normalized gradient direction vectors [TLF10], [Li14]. Winder *et al.* [WHB09] showed how to learn the optimal DAISY descriptors. Their experimental work showed that the correct DAISY descriptor is superior in comparison to e.g. SIFT and SURF. Marimon *et al.* [MBAG10] proposed the DART descriptor, a optimized DAISY descriptor with 8 direction gradients and 2 layers of central-symmetrical circles.

2.4.1.4 Binary features

Feature performance, including detection, description, and matching speed is important to reach real time performance in object recognition. In order to make this process as fast as possible it is important that features consume as littel memory as possible. One way is to reduce the dimensions of the feature descriptor by PCA or similar. Binary features is a way to speed up matching and reduce memory consumption by directly condensing image patches without computing a descriptor. A review and comparison of local binary features is presented in [HDF12].

The BRIEF descriptor [CLSF10] was one of the first binary features. It describes small smoothed image patches as binary strings. The binary descriptor is created from test responses where pairs of pixel in the selected image patch are compared in a brightness tests. If the intensity of $p(x) < p(y)$ then the test response is 1 and 0 if not. The number of tests are typically 128, 256 or 512, which corresponds to a 16, 32 or 64 bytes of memory used. As a comparison, a SIFT feature is a 128 dimensional feature stored as floating point numbers, which take up 512 bytes. Another benefit of binary features is that the feature comparison metric during matching is the Hamming distance, which basically is a XOR operation instead of the computational expensive L_2 norm. A shortcoming of BRIEF is that it is not rotation and scale invariant. Leutenegger *et al.* [LCS11] proposed the "Binary Robust Invariant Scalable Keypoints" (BRISK) descriptor, which provides scale and rotation invariant. It computes interest points using a modified AGAST as described earlier. Opposite to the BRIEF descriptor, BRISK uses a symmetric sampling pattern to generate test responses. The sampling pattern is similar to the DAISY descriptor and is build of non-overlapping concentric circles. A BRISK descriptor string is 512 bits equal 64 bytes. Rublee *et al.* [RRKB11] proposed the "Oriented FAST and Rotated BRIEF" feature to introduce a FAST like detector that is both scale and rotation invariant by adding a scale-space similar to BRISK and a orientation component similar to SIFT/SURF. ORB was developed to get free from the licensing restrictions of SIFT and SURF, and is free to use in e.g. OpenCV. Alahi *et al.* [AOV12] introduced the "Fast Retina Keypoint"(FREAK), which is inspired by the human visual system. FREAK uses concentric circles as sampling pattern, which is overlapping in opposition to BRISK. FREAK uses an exponentially point pair sampling strategy by sampling more points in the inner circles. Unsupervised learning is used to choose an optimal set of point pairs under the restriction of exponentially sampling.

The research activities in Binary features has started to accelerate within the

last five years. The presented features, BRIEF, ORB, BRISK and FREAK are the most established. Newly proposed and upcoming binary features like LDB [YC14], M-LDB [PAGIoT13], DBRIEF [TL12], LDA-Hash [CSF12] and LATCH [LH15] are not discussed in this review.

Feature comparative studies:
Aanæs *et al.* [ADSP12] presented a comparative study that investigated the stability of 10 common interest point detectors with respect to large changes in viewpoint, scale, and lighting. Their evaluation are based on their own dataset taken with an industrial robot. The dataset consists of 60 scenes with precise ground truth taken with structured light and controlled light settings. The conclusion of this study was that the three scale-space detectors, Harris corner, Hessian Blob and Difference-of-Gaussian are preforming best. Recently, Mukherjee *et al.* [MJWW15] presented an evaluation of interest point detectors and feature descriptors that included some of the recent proposed interest point detectors like BRISK, BRIEF, FREAK, CenSurE, ORB, SIFT, SURF and some common with Aanæs *et al.* [ADSP12]; FAST and MSER. Their study showed a high performance of the SIFT feature and good performance of some of the newer interest point and feature descriptor like ORB, BRIEF and FAST. Several evaluations of interest point detectors targeting different application areas like Visual SLAM [GMBR10] and Pose Estimation [VFJM09] are proposed. During the years of research in interest points and feature descriptors, several studies have compared the performance of different combinations of interest point detectors and feature descriptors [MP05], [DAP11], [KTF11].

2.5 Feature Matching

Once features are extracted from a test image and their descriptor computed, the next step is to establish correspondences between the feature set of the test image and the feature set of a model e.g. one or several images of an object. To establish correspondences a metric and a search strategy for finding the nearest neighbours are needed. The task in nearest neighbour search also known as similarity search is to find similar features by searching a higher dimensional space. In order to reduce the dimensionality of the feature search space, which is 128 for the SIFT features; the L_2 norm / euclidean distance is typically used as similarity metric in feature space. Other metrics like L_1 and L_∞ are alternatives but in general the L_2 yields the best results. In case of nearest neighbour search for binary features, metric like the Hamming distance is used [ML12].
When searching for correspondences, a metric for measuring the performance or the rate of correct matches is required. In object recognition, pose estima-

tion and other literature that want to measure the performance of matching strategies, the terms ***true positive***, ***false positives***, ***false negatives*** and ***true negative*** are measures for the matching and/or recognition performance. ***true positives*** is the number of correct matches where ***false positives*** are estimated matches that are incorrect. The fact that matching algorithms have ***true positives*** you can also see ***false negatives***, which are matches that are not correctly detected. Opposite, ***true negatives*** are non-matches that are correctly rejected. With these terms we can establish two performance measurements; precision and recall defined in Equation 2.5 and Equation 2.6, respectively.

$$precision = \frac{true\,positive}{true\,positive + false\,positive} \tag{2.5}$$

$$recall = \frac{true\,positive}{true\,positive + false\,negative} \tag{2.6}$$

The nearest neighbour search problem is defined as follows: given a finite point set $P = p_1, ..., p_n$ in a k dimensional vector space X, the algorithm must preprocess the point set P such that the nearest point in P to a query point $q \in X$ is found in an computational efficient way. The simplest methods for correspondence search strategy is to make a brute force search by setting a threshold on the max euclidean distance and return all matches. However, this strategy is computational expensive for large feature sets and not very precise because the threshold parameter needs to be adjusted each time a new object is taken into account. A better strategy is to adapt an indexing structure such as a multidimensional search tree or a hash table for rapid similarity search.

2.5.0.1 Partitioning Trees

A widely used search strategy is multi-dimensional search trees where the best known is the Kd-tree (*k-dimensional tree*), which was initial proposed by Bentley *et al.* (1975) [Ben75] and Friedman *et al.* (1977) [FBF77]. A kd-tree is a nearest neighbour search algorithm, which takes a finite point set as input and create a k-dimensional tree where each node is a k-dimensional point. Kd-trees are binary search trees and a space-partitioning algorithm that splits all children nodes along a specific dimension that exhibits the greatest variance. At the root level the point set is split into two halves; one greater and one smaller than the root in the specific splitting dimension e.g. the x direction. Typically the median point of the finite point set is selected as root. This procedure is repeated for every child in the tree. Finding the nearest neighbour to a query

point in a constructed kd-tree is conducted by recursively moving down the tree and check whether any child points of the current node are closer to the query point than the current best. If that is the case, then move to that node and continue the search until the closed point in that branch is found, see Figure 2.11 (a)-(c). When the closes point is located, a hypersphere around the point with a radius equal the distance to the closest point are constructed in order to find additional closest points, Figure 2.11 (d). If the hypersphere crosses a hyperplane there could potential exist additional points, which is closer to the query point. In order to serch in other leafs, backtracking is performed where the neighbouring nodes are unwind to search for closer points and a new hypersphere is constructed at the new closest point, Figure 2.11 (e)-(g). If the hypersphere intersect with the root hyperplane or other hyperplanes, see Figure 2.11 (g), the tree is traversed to the root to be able to examine the other root branch, Figure 2.11 (h)-(l). The other branch of the root node is then traversed in a similar way to find points closer than the current closest point, Figure 2.11 (m)-(t).

Kd-trees are very efficient in low dimensionality spaces to find the exact nearest neighbour but the performance quickly decreases for high dimensional data. In worst case the entire tree has to be traversed to find the nearest neighbour, which has a computational complexity of O(log n). In order to make efficient search the algorithm can be extended to find the k nearest neighbours to a query point instead of just one. "A branch in the tree is only eliminated when k points have been found and the branch cannot have points closer than any of the k current bests". The kd-tree algorithm can further be converted to an approximated nearest neighbour search algorithm [BL97], [AMN+98], [ML09], which aim at finding the nearest neighbour fast by setting a constrain on the search. Approximated nearest neighbour search provides large speed-ups with only minor loss in accuracy. Two of the most established ANN search strategies priority search, which are using a distance threshold for the nearest neighbour search [AMN+98] or setting a fixed number or time of points to visit [BL97]. The method proposed by (Beis and Lowe) [BL97] known as the Best Bin First algorithm set a restriction of the number E_{max} of nodes to visit and in addition the algorithm examines only the bins or partitioned spaces with the lowest distance to the query point during backtracking. This is implemented with a simple priority queue that stores the distance to the query point and the current tree position. After a branch has been examined the top entry in in the priority queue is removed. The small improvements enable the BBF algorithm to find the nearest neighbour point must faster, especially when the data points and dimension increases, and the algorithm finds a larger fraction of correct NN compared to the conventional kd-tree.
Multiple randomized kd-tree [SAH08] is another method for approximated nearest neighbour search, which constructs m different oriented trees each with a different and largely independent structure. During search, m different trees are

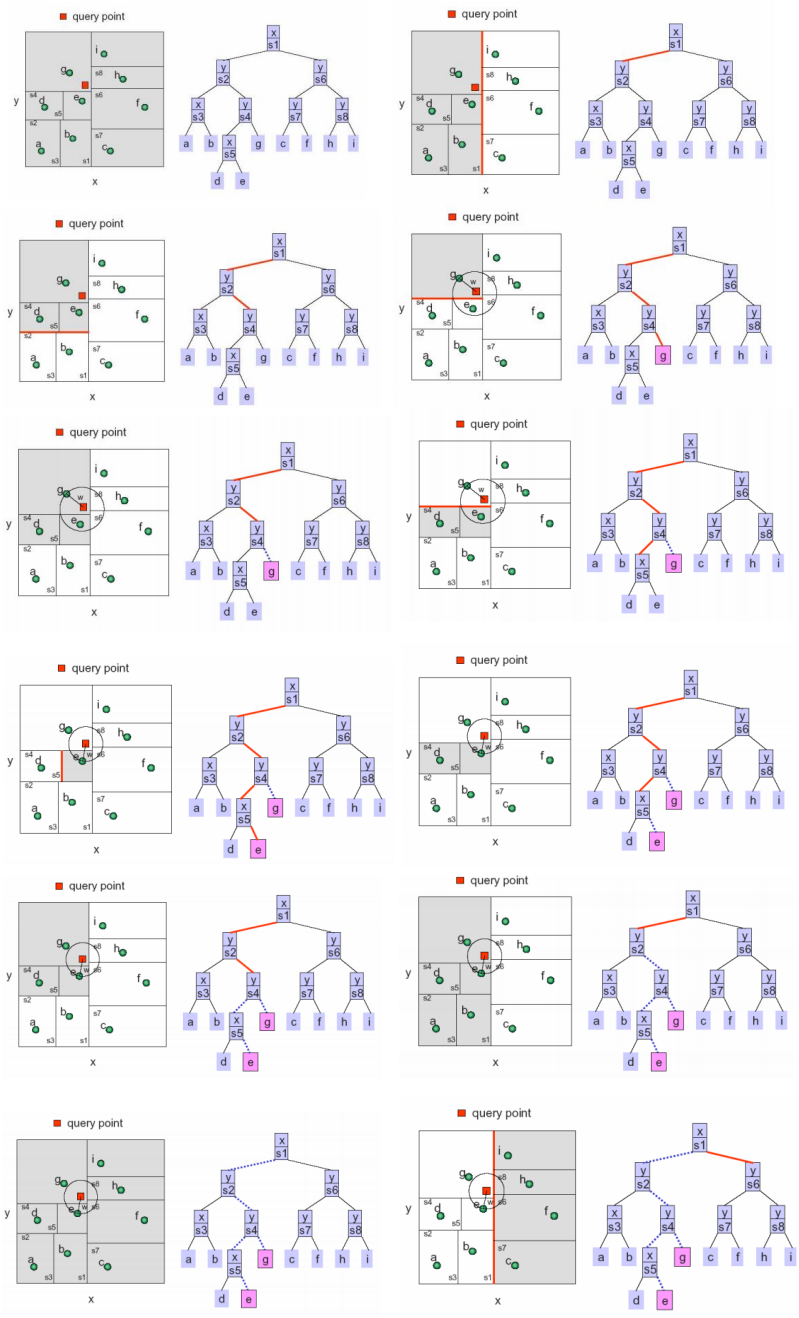


Figure 2.11: Spatial decomposition of a two-dimensional space with kd-tree. `https://www.cs.umd.edu/class/spring2008/cmsc420/L19.kd-trees.pdf`

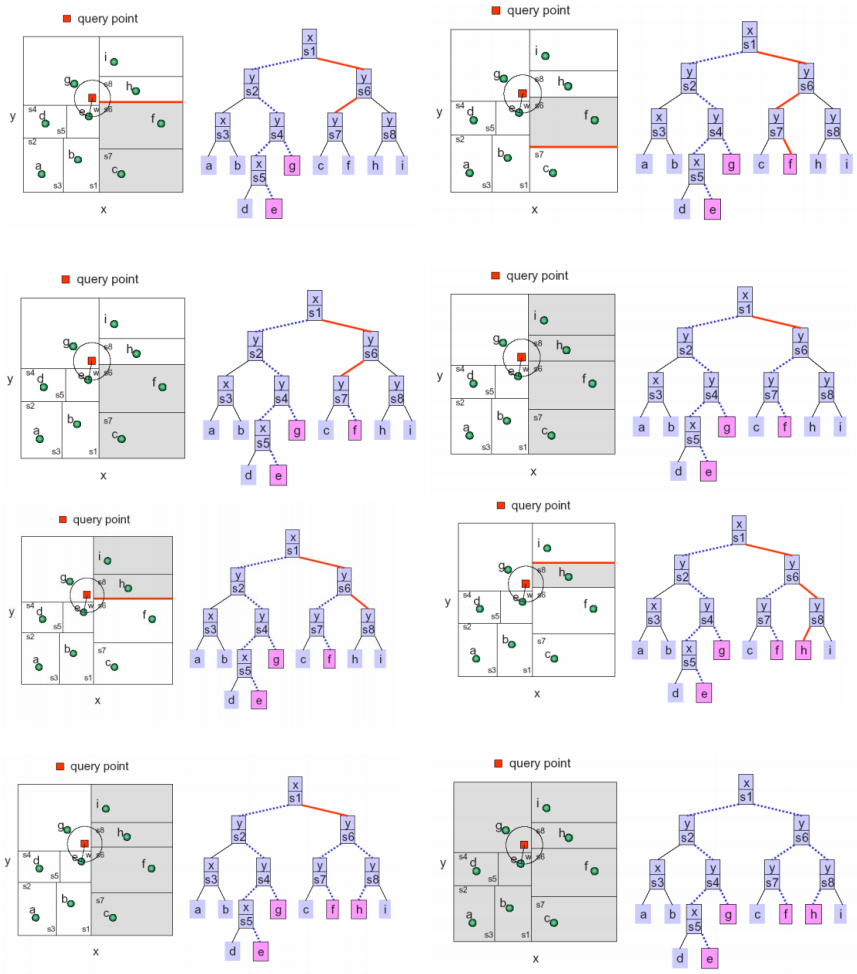


Figure 2.12: Spatial decomposition of a two-dimensional space with kd-tree. `https://www.cs.umd.edu/class/spring2008/cmsc420/L19.kd-trees.pdf`

searched simultaneous with only n nodes visited in each tree. This results in an average search of n/m nodes per tree. After trees are traversed once the backtracking uses a shared priority queue to find the global nearest neighbour. This method is refered to as NKD-Trees. Other, approaches of randomized kd-tree include RKD-Trees with randomly chosen split points and dimension but without rotating the data as with NKD-Trees and PKD-Tree, which is align the data with the principal axis from PCA. All, methods are proposed by Silpa-Anan and Hartley [SAH08] and they concluded that the PKD-Tree preformed the best. In [ML09] a wide range comparison supported that multiple randomized k-d-trees are one of the most efficient matching algorithms.
Another class of partitioning trees are hierarchical k-mean kd-trees [FN75] that decompose the space with a clustering algorithm instead of using hyperplanes. The most well-known k-mean ANN is the vocabulary tree proposed by (Nistêr and Stewênius) [NS06], which is building a search tree by clustering the data and defining k cluster centers in the training phase. The training is partitioned into k groups consisting of the point closest to the cluster center. This is then recursively applied to each group, which are split into k new quantization cells. This continues until the maximum of L levels are reached. The process is illustrated in Figure 2.13. In the online nearest neighbor search, a query descriptor is propagated down the tree and compared at each level to the k candidate cluster centers and and the closest one is selected. The dot product is used to compare the two feature vectors at each level, resulting in kL dot product computations, which is very efficient if k is not too large. Thus, the Vocabulary trees is very suited for large database search. Other examples of decomposing the space via clustering includes GNAT [Bri95], the anchors hierarchy [Moo00], the vp-tree [Yia93], the cover-tree [BKL06] and the spill-tree [LMYG04].
Many of the nearest neighbour search algorithms are released in the open source software library named Fast Library for Approximate nearest neighbors (FLANN)[10] by Marius Muja [ML09]. Recently, (Muja and Lowe) [ML14] investigated the performance of various algorithms and found that the multiple randomized k-d tree [SAH08] and their newly proposed priority search k-means tree [ML14] preformed the best.

[10] FLANN: `http://www.cs.ubc.ca/research/flann/`

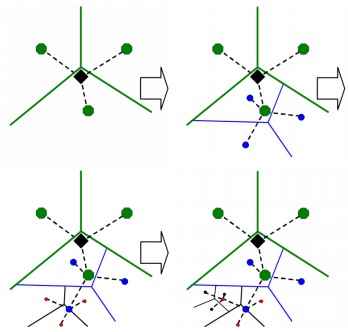


Figure 2.13: In a vocabulary trees, search tree are build by clustering the data. [NS06]

2.6 Robust estimation

Together with the feature description and matching, a robust estimation step is part of every pose estimation pipeline. In general the task of Robust estimation is defined as:

"Estimation techniques in computer vision applications must estimate accurate model parameters despite small-scale noise in the data, occasional large-scale measurement errors (outliers), and measurements from multiple populations in the same data set.", [Ste99]

In pose estimation application the concept of robust estimation covers methods to remove as many wrong feature matches as possible. Once hypothetical corresponding matches are established, we can often use geometric alignment to verify which matches are correct and which ones are failures. This is often referred to as inliers and outliers. Robust estimation techniques are divided into a two-stage process. First, classify data points as outliers or inliers and secondly, fit a mathematical model to the inliers while ignoring outliers. However, some methods are iteratively running through the two steps e.g. RANSAC. There are two popular methods to determine outliers, the RANSAC algorithm and M-estimators. Despite of the effectiveness of both algorithms, only RANSAC will be covered because of the popularity.

2.6.1 RANSAC

The RANSAC algorithm was original proposed as a robust estimation technique in a Prespective-3-Point (P3P) pose estimation problem [FB81]. The concept of P3P will be covered in Section 2.6.2. Since the original paper was published RANSAC has been applied to many computer vision problems such as PnP, visual SLAM, homography estimation, fundamental or essential matrix estimation, etc. RANSAC is a iterative hypothesis and test algorithm to determine inliers in a point set. The algorithm randomly select a small sunset of corresponding points to generate a pose hypothesis. For each hypothesis a PnP algorithm estimates a pose hypothesis, which is used to compute the reprojection error. Those point where the reprojection of the world points is closer than an threshold to the image points is categorize as inliers. The RANSAC step is:

1. Random select a minimum of correspondences S_k (e.g. 3 for P3P and 4 for P4P algorithms)

2. Compute the pose $[R|t]$ from these minimal set of correspondences using a Perspective-n-Point algorithm, see Section 2.6.2

3. Determine the number of inliers from the whole point set of correspondences

4. Repeat step 1 to 3 until convergence criteria is meet

After all inliers are resolved using RANSAC, a more accurate PnP approach like an iterative algorithm, which considers all the determined inliers can be applied as a pose refinement step. The number of required RANSAC iterations to ensures a probability p that at least one sample with only inliers is drawn can be determined automatically with Equation

$$N = \frac{log(1-p)}{log(1-(1-\eta)^n)} \tag{2.7}$$

where η is the probability that a correspondence is an outlier, p is the probability that at least one sample with only inliers is selected and n is the data size per sample. For P4P problems $n = 4$. With this formula 5 iterations are required if 10% of the correspondences are outliers and $p = 0.99$ and 72 iterations when 50% are outliers.

2.6.2 Perspective-N-Point

In classic single camera pose estimation we want to determine the pose of an object by considering the 2D image projection of minimum three 3D world points. The problem is known as the Perspective-N-Point problem - PnP. The aim is to determine the camera pose given its intrinsic parameters and a set of n correspondences between 3D points on the real world object and their 2D projection. In this review, only Perspective-n-Point algorithms for solving the absolute pose problem for central cameras are investigated. Methods to estimate the relative pose between camera frames by considering 2D-2D correspondences, like the 8-point algorithm [HZ04], [Har97] are not considered because we investigate the problem of object pose estimation from a single camera. Perspective-N-Point algorithms for non-central cameras with unconstrained projection rays where the rays of lights are not centred in the center of projection of the camera [SRT+11] is not reviewed in this thesis. The geometric configuration of the four different problems are illustrated in Figure 2.14.

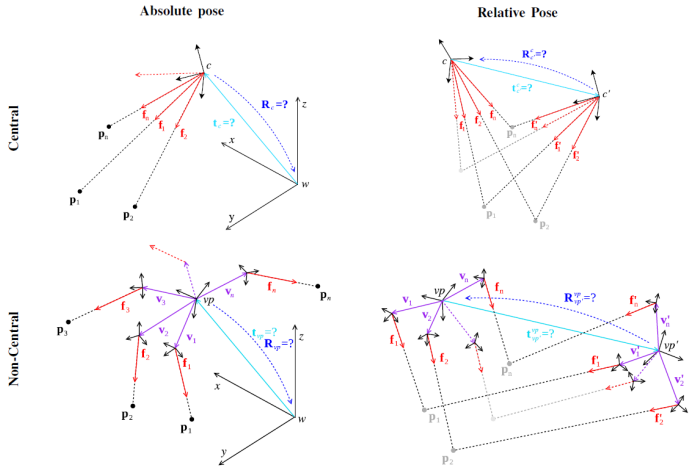


Figure 2.14: Different types of Perspective-n-Point problems. **Upper left:** Absolute pose for central cameras **Upper right:** Relative pose for central cameras. **Lower left:** Absolute pose for non-central cameras. **Lower right:** Relative pose for non-central cameras Kneip2014

During the last three decades this topic has been relevant as an ongoing research topic because of the many application areas, especially within robotics. Some

of the application areas in robotic count pose estimation of objects in industry, robot navigation and visual odometry, camera tracking, camera calibration, 3D estimation with structure from motion and many more. The problem of the Perspective-n-Point pose estimation can be classified into iterative and non-iterative solutions. The solutions proposed are able to solve the camera pose problem by applying down to three non-collinear corresponding 3D/2D point pairs. In general, both the iterative and non-iterative methods have pros and cons. The iterative algorithms are more accurate than the non-iterative ones but comes with high computational cost and the risk of instability due to local minima of the cost function [LXX12]. The iterative algorithms are typical numerical stable but some of the algorithms need an initial guess of the camera pose to converge. The non-iterative algorithms are fast and efficient because the pose estimation problem can be solved in closed form, but the algorithms have instability in the presence of noise especially when the 3D world points are limited, $n \geq 5$. In general more 3D/2D correspondences will increase the accuracy of the algorithm. When we have $n \geq 4$ points, the solution is in general considered unique. Some dedicated algorithms exist for solving the P4P problem e.g. [HW02], which presented an analysis of the probability of more solutions.

Direct Linear Transformation

The Direct Linear Transform (DLT) is the most straight forward method for recovering the pose. It is considered as the starting point of pose recovering algorithms [AAK71], [HZ04]. The Direct Linear Transform algorithm is a linear function, which maps P_{world} to p_{image} and estimates the projection matrix $\mathbf{P}$ by solving a linear system of equations with a minimum of 6 correspondences. The DLT algorithm is known to achieve relatively accurate results from a large number of points, [LXX12]. However, with few points the Direct Linear Transform method is quite inaccurate due to overlooking the known calibration parameters. Equation 2.8 and 2.9 show the basic DLT Equation;

$$\frac{P_{11}X_i + P_{12}Y_i + P_{13}Z_i + P_{14}}{P_{31}X_i + P_{32}Y_i + P_{33}Z_i + P_{34}} = u_i \tag{2.8}$$

$$\frac{P_{21}X_i + P_{22}Y_i + P_{23}Z_i + P_{24}}{P_{31}X_i + P_{32}Y_i + P_{33}Z_i + P_{34}} = v_i \tag{2.9}$$

where P_{ij} is the projection matrix, u_i, v_i are the i^{th} 2D image points and X_i, Y_i, Z_i are the i^{th} 3D world points. The derivation of the two Equations

are found here[11] and a Matlab implementation here[12] Equation 2.8 and 2.9 can be expressed in matrix form as Equation 2.10

$$Ap = 0 \tag{2.10}$$

where $\mathbf{p}$ is a vector composed by the coefficients $\mathbf{P_{ij}}$. The solution to this linear homogeneous equation can be found from the Singular Value Decomposition (SVD) of $\mathbf{A}$ and taking the eigenvector with the minimal eigenvalue. The camera pose is extracted from the projection matrix $\mathbf{P}$ with $[R|t] = K^{-1}P$.

P3P - *A special case of the Perspective-N-Point problem*
Determine the camera pose is theoretical possible with only 3 3D/2D correspondences, because it is possible to represent a full pose with only 6 numbers(x,y,z,roll,pitch,yaw). This specific problem is known to have up to four different solutions if no precaution is taken as a post computation step where additional information is required to guarantee the uniqueness of the solution. This fact is typical referred to as the fourfold ambiguity [WMSM91]. Most P3P algorithms are using two steps in order to estimate the pose. First step is to solve the projection of the 3D world points ($\mathbf{A}$,$\mathbf{B}$,$\mathbf{C}$) into the camera image plane ($\mathbf{v}$,$\mathbf{u}$,$\mathbf{w}$) in Figure 2.15 in the camera reference frame. Thanks to the constrain given by the three triangles (P_{cam}, P_A, P_B), (P_{cam}, P_A, P_C), (P_{cam}, P_B, P_C), where P_{cam} is the center of the projection and P_A, P_B, P_C are the 3D world point pairs, we can apply the law of cosine to each of the triangles. This estimates the unknown depth Z_i by solving a fourth order polynomial equation [FB81],[ATQ00],[GHTC03]. Now, the second step is applied to compute the transformation $[R|t]$ of the camera by aligning the projected 3D world points with the 2D image points. The transformation is found in closed-form solution using quaternions [Hor87] or singular value decomposition (SVD) [AHB87], [Ume91]. This problem is also known as the Absolute orientation problem.

Haralick *et al.* [HLON] presented in their review paper the early work on all the major direct solutions before 1991. Newer proposed work includes [QL99], [ATQ00], [GHTC03],[KSS11]. Recently, Kneip *et al.* [KSS11] proposed a novel closed form solution, that estimates the camera pose directly in a single step, instead of projecting the 3D points to the image plane and then align the points as previous methods. The method introduces a new intermediate camera frame in the center of projection of the camera whose x-axes is aligned with the first

[11] `http://www.kwon3d.com/theory/dlt/dlt.html`
[12] `https://www.mathworks.com/matlabcentral/fileexchange/47032-camera-geometry-algorithms/content/CV/CameraGeometry/DirectLinearTransformation.m`

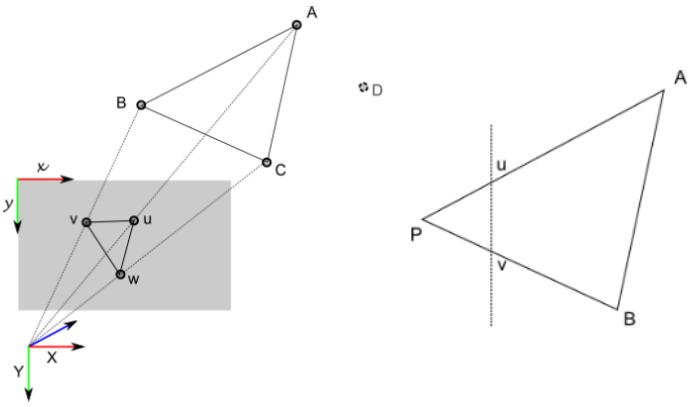


Figure 2.15: The P3P pose estimation problem

world point P_A vector and secondly a new world frame, which is centered in P_A whose x-axes is aligned with the direction of the second point P_B. The relative transformation between the two frames can be represented using only two parameters and are estimated by solving a fourth order polynomial equation. A final substitution allows computing the camera pose. The algorithm has a low computational cost and is faster than other P3P solutions because it runs in a single step. The estimation of the camera pose is computed in around 1.5ms at a standard laptop computer. The algorithm is avaliable in OpenGV [KF14]. The fact that P3P algorithms suffer from the fourfold ambiguity, often a fourth point P_D and its corresponding image point Z_D are used to be able to find the best solution among the four solutions. In practice this makes the P3P algorithm to a P4P solution. However, applying P3P pose estimation algorithms are often an obvious solution to bootstrap Perspective-n-Point non-linear optimization methods that minimizes the re-projection error, as a prior guess before minimization of an objective function.

PnP

In the situation where $n \geq 4$ it is possible to compute a unique solution. A straight forward solution to the PnP problem is to compute the depth of the points first and then retrieve the 3D world coordinates in the camera frame. Then it turns into a well-known 3D-3D relative pose problem, where it is simple to compute the transformation that aligns the 2D image points and the projected 3D points using quaternions [Hor87] or singular value decomposition (SVD) [AHB87], [Ume91] in the world frame. During the years many differ-

ent approaches have been proposed with the aim of solving the Perspective-n-Point problem by low computational cost and pose accuracy, non-iterative methods [DRLR89], [HCLL89], [QL99], [AD03], [HW02], [LFNP09], [LXX12], [LXX12],[ZSO13], [KFS13], [KF14] and iterative methods [ODD96], [DD95], [LHM00], [OKO09], [GCF12]. In general iterative methods are considered to be slow compared to the non-iterative.

The method by Triggs *et al.* [Tri99] generalizes the 6D Direct Linear Transform (DLT)[HZ04] by incorporating prior camera knowledge (intrinsics). The minimum number of 3D world points required is reduced to 4 or 5 compared to DLT which requires $n \geq 6$, where the 4 point method recover focal length and the 5-point method recovers the focal length and principal point. This method does not perform well for large number of points as pointed out in [AD03]. The algorithm from [Tri99] together with,[AC95] are one of the few Perspective-n-Point algorithms for un-calibrated cameras. Abidi and Chandra [AC95] proposed a solution for the coplanar 4-point configuration that estimates the pose and the unknown focal length for a perspective camera. Even if four 3D-2D correspondences are adequate, it is nonetheless desirable to have a large set of correspondences to introduce redundancy, which increase the accuracy and lower the sensitivity to noise. Typically, an outlier rejection method like RANSAC [FB81] is desired to get a robust estimation. Quan and Lan [QL99] presented two algorithms; a direct linear 4-point and a two step 5-point, which can be extended to n. The algorithms consider triplets of world points to estimate the unknown depth Z_i, by solving four-degree polynomial. This homogeneous linear equation is solved using SVD. The major problem with the two algorithms are the computational complexity, which is $O(n^5)$. This was improved by Fiore [Fio01] that proposed a non-iterative algorithm with a complexity of $O(n^2)$. It has later been shown by Asar *et al.* [AD03] that the solution is noisy for unstable 2D. Asar *et al.* [AD03] proposed an linear algorithm for both n points and n lines, which solves the problem but with a much higher computational complexity, $O(n^8)$ as a consequence.

In 2008, Lepetit *et al.* [LFNP09] introduced the EPnP algorithm, which was the first non-iterative solution with an $O(n)$ computational complexity. The low computational complexity is meet by introducing four non-coplanar virtual control points to represent n 3D world points as a weighted sum of the null eigenvectors. The four virtual control points are selected by taking the centroid of the n 3D world points as one and select the rest such that they form a basis, aligned with the principle directions of the data. This reduces the estimation problem to estimate the coordinates of the control points in the camera frame. The EPnP algorithm gives valid pose estimates for planar point configurations by selecting 3 virtual points instead of 4, which is the case for the general

configuration. As discussed by Li *et al.* [LXX12] the EPnP suffering from low accuracy for slightly redundant cases with n=4 or n=5.

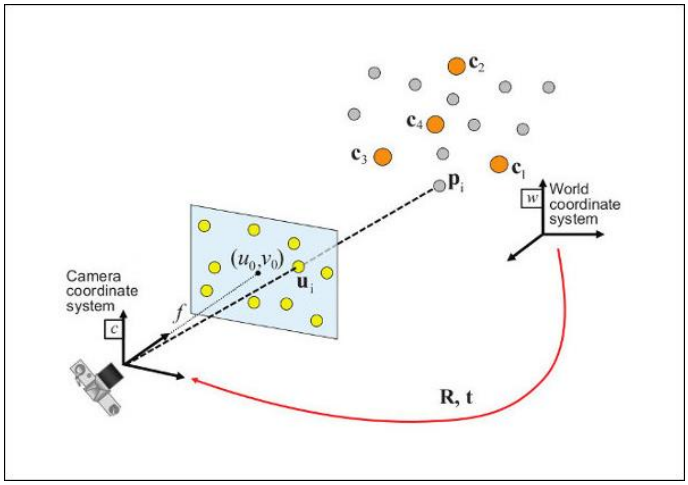


Figure 2.16: Principle of the EPnP algorithm. `http://docs.opencv.org/3.1.0/dc/d2c/tutorial_real_time_pose.html`

With the significant contribution from Lepetit *et al.* [LFNP09], recently new method based on these ideas have been proposed that all aim at linear complexity. Li *et al.* [LXX12] proposed the Robust-PnP,(RPnP) that improved the accuracy compared to EPnP. Previous methods have applied a two-step methodology to the pose estimation problem, which estimates the depth and then transform the problem into a 3D-3D registration problem. "The Direct-Least-Squares (DLS) method [HR11] formulates a nonlinear cost function, that produces a fourth order polynomial system and is solved using the Macauley matrix method. The main drawback in DLS is related to the Cayley representation used for the rotations, which is degenerated at 180 degrees", [FBMN14]. The DLS method has a computational complexity of $O(n)$. The work from Zheng *et al.* [ZSO13], [ZKS$^+$13] proposed two direct minimization methods by parametrizing the rotation as a non-unit quaternion solved by means of a Gröbner basis solver; the Accurate and Scalable PnP (ASPnP) and the Optimal-PnP (OPnP). The solutions is scalable in means of large number of n and more accurate compared to EPnP and RPnP [ZKS$^+$13]. It is reported that ASPnP is more than 5 times faster than EPnP and RPnP with $n = 5$. Lastly, Kneip *et al.* [KF14] proposed the Unified-PnP, UPnP, which is the first non-iterative universal solution with linear complexity that is complete in all its properties. It includes pose estimation for both central and non-central cameras, where other

methods like [ZKS+13] only are valid for central cameras. The UPnP algorithm is faster compared to previous methods. In a typical pose estimation pipeline outlier rejection is a separate process where pose hypothesis are tested in a geometric verification step. Ferraz *et al.* [FBMN14] recently integrated the outlier rejection directly in the pose estimation step and gained 100 x speed compared to a conventional RANSAC pipeline.

Iterative methods:
In contrast to closed form solutions, direct iterative minimization methods minimize a defined error function in the image or object space. The iterative methods take all nonlinear constraints into consideration. The error function to minimize is typically the re-projection error, which is widely recognized as the best criterion [ZKS+13] but some methods uses an algebraic error [Har98] e.g. [LHM00]. The method proposed by Lu *et al.* [LHM00] developed an orthogonal iteration method, which directly minimizes the object space error. The method from Garro *et al.* [GCF12] offered an alternating minimization method to minimize an algebraic error defined in the image space, [ZKS+13]. One of the benefit of choosing an iterative methods instead of a non-iterative is cases where few or non redundant points are available. In these cases iterative methods provide more accurate results than the non-iterative, [LXX12]. Some of the drawbacks of iterative methods are that they are computational expensive and many algorithms need a prior pose hypothesis. Furthermore, the methods suffer from the potential instability due to the local minima of the cost functions. Schweighofer and Pinz [SP08] proposed their SDP method, which partially addressed the problem of multiple local minima in the case of coplanar point sets. Their method uses semidefinite programme (SDP) to solve the PnP problem. Olsson *et al.* [OKO09] proposed a branch-and-bound to be able to compute the global optimum but with high computational cost as a consequence. The third major shortcoming of iterative methods is that some of the algorithms [LHM00],[SP08],[OKO09],[GCF12] only return a single solution, which might not correspond to the correct pose in case of multiple solutions, [ZKS+13].

The method proposed in [DD95] (POSIT) applies iteratively a linear closed-form solver. POSIT uses a scaled orthographic projection (SOP), which leads to a linear system of equations. In concept, the scaled orthographic projection creates a virtual plane where the 3D world point is projected. From the SOP a coarse pose estimate is computed, which is then iteratively improved until a reprojection error tolerance is met. The POSIT algorithm requires $n \geq 4$ and no initial pose. The computational complexity is typical $O(n)$ but the method suffers from the fact that in applications with low focal lehgths or a object is close to the camera, the SOP assumption is not a valid apporximation, which leads to unaccurate results [SD16]. The method [DD95] is not considering coplanar points. An extension to POSIT has been proposed in [ODD96], which enables pose estimation form coplanar point sets. This implementation is available in

OpenCV and VISP [13] [MSC05].

Algorithms for solving the Perspective-n-Points problem are available in OpenCV, VISP [MSC05], OpenGV [KF14]and OpenTL [PLW+08] among others.

2.7 Related implementations

In this section significant related work on feature based object detection and 6 degree-of-freedom pose estimation will be presented. The focus is systems for robotic guidance that apply the detection pipeline presented in the previous sections or modifications of this.
The work done by David Lowe, 1999 [Low99], is considered as the baseline for feature based object detection. He showed how to apply the newly invented SIFT feature for scale invariant object detection of textured objects. Objects are detected in the image by extracting SIFT features from a single training model image and in a test image. Features are matched by a kd-tree nearest neighbour search, the best bin first method. Pose hypotheses are clustered using the Hough transform and an affine transformation is computed, in a least-square manner, to make geometric verification. The methods could readily be extended to estimate a 3D pose by applying a perspective-n-Point algorithm.

Feature based learning of world models:
Gordon And Lowe [GL06] extended this pipeline by building a sparse 3D model of local features instead of a model only consisting features from one view. In their approach, 3D SIFT feature models are created from 5 to 20 images taken of an object from different views with a handheld camera. From the acquired image set, the two images that are spatial closest to each other are found by feature matching with the Best bin first algorithm. Outliers from matches are removed by applying an epipolar geometry constraint and the fundamental matrix is computed. This step is repeated until a circle is created in the view tree. 3D coordinates of the sparse feature model are computed using iterative bundle adjustment using a Levenberg-Marquardt optimization algorithm. Now online pose estimation is a matter of matching 2D-3D correspondences between image and model points with a BBF algorithm, compute the pose with an iterative algorithm that minimizes the reprojection error and filter outliers by checking the geometric consistency with the RANSAC algorithm. During the work from Carnegie Mellon University (CMU) making the HERB robot grasping objects autonomously, Collet *et al.* [CBSF09],[CS10],[CMS11] presented the MOPED framework, "**M**ultiple **O**bject **P**ose **E**stimation and **D**etection". They presented

[13] `https://visp.inria.fr/`

a full system for modeling and recognizing multiple objects in a domestic setting. Sparse 3D models are learned by extracting SIFT features and applying structure from motion to spatial orient the features into a 3D feature model from training images. Nakada *et al.* [NKM10], [TKM11] showed the same pipeline to create "SIFT-Cloud-Models" but accelerated in a GPGPU.

With the introduction of the Kinect sensor in 2010, new methods for learning spare feature model without the use of structure from motion techniques were introduced. One of the major problems before 2010 was that recognition systems relied on offline object learning in a controlled environment, in order to segment the object from the background. Zillich *et al.* [ZPMV11] used a Kinect like sensor to segment an object and the table in the 3D point cloud, and tracking the object in image space by extracting SURF features. New key frames are added to the model by considering the amount of unseen 2D features. Pangercic *et al.* [PHB11] proposed a similar idea in the construction of a generic perception module for the PR2 domestic robot. The ODUfinder, "Objects of Daily Use Finder" recognizes objects using SIFT features and a vocabulary tree for search in an object database. If novel objects are detected the robot grasps the object, presents the object for the RGB-D camera and rotate the object. SIFT features are detected, a new document is created for the vocabulary tree and stored in the database. In the last decade autonomously learning of object models for recognition and pose estimation tasks have increasing interest; especially in humanoid robotic domain. Research is still ongoing in building spare feature models of object. However, with the introduction of the Kinect the trend goes toward building and learning dense point cloud models directly in 3D.

Rothganger *et al.* [RLSP06], [KP06] extended the sparse 3D feature model to contain affine invariant regions. Their object model consist of small affine rectified image patches shaped as parallelograms which are spatial separated, taken from camera views covering 360 degrees. Each patch is invariant to changes in illumination by normalization of the patches. The modelling framework locates interest points with the Harris and Difference of Gaussian and matching surface patches using the SIFT descriptor. Correct corresponding matches between surface patches are found using k-nearest neighbour with the euclidean L2 norm, followed by comparing patches with normalized correlation. Additional, outliers are removed by exploiting the epipolar constraint between two images and with RANSAC to check the geometric consistency. Each paired images stitched together and the 3D structure is optimized with bundle adjustment. In [DPP09] hierarchical visual primitives computed from a stereo camera are temporal accumulated and integrated in a probalistic framework, by extracting small image patches along object contours while a robot rotates an object in the camera frustum. These early cognitive vision (ECV) features are inspired by the human visual system and the processing pipeline has a similar hierarchical structure where image features like edges, junctions, edge color and optical

flow are combined to 2D primitives. These 2D primitives are reconstructed and become 3D primitives [PK11]. The accumulation of 3D primitives is possible with a single camera but with lower accuracy.

Multiple views - a method for increasing precision:
In order to gain a required precision in robotic applications it is crucial that 2D feature points are estimated accurately. In applications such as domestic robotic or other applications with scene mounted cameras where the main purpose of the camera is to give an overview of the scene, pose estimation from 3D-2D correspondences can lack the needed pose accuracy. In these cases robot mounted cameras are required in order to move closer or simply use several cameras with different field of views. If this is not applicable other methods like 2.5D or full 3D pose estimation can be necessary where some kind of 3D sensor is used. These topics are discussed in Chapter 4 Another, yet simple approach is to use more than one camera to increase the accuracy of the pose estimate. Several approaches has been reported in the past that increase the accuracy and performance using multiple views. Viksten *et al.* [VSNP06] presented how integration of several pose estimation algorithms using complementary features increased the accuracy and robustness towards illumination changes. Furthermore, it showed how pose estimation results from different views of the scene increase the accuracy. Azad *et al.* [AAD07] used a stereo camera and the epipolar constrain to locate shi-tomasi features detected in left camera in the right camera to compute accurate depth of textured box objects before grasping with a humanoid robot. The same author compared this stereo based pose estimation method with conventional monocular pose estimation methods based on 2D/3D correspondences and found that applying stereo camera significantly increased the depth accuracy of the pose estimate [AAD09]. Grunmann *et al.* [GES+10] showed a similar approach but instead of considering box object their algorithm worked for arbitrary object geometry. The author that proposed the Moped framework extended their work to multiple views in order to increase the accuracy of the MOPED system [CS10]. Their approach estimated the pose of objects using 2D/3D correspondences in each monocular view and optimized the global pose hypothesis by clustering in pose space and minimizing the reprojection error.

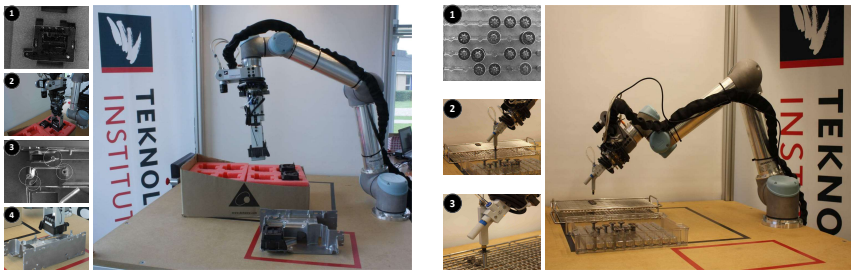
2.8 Robot Skills - An enabler for generic vision components

In this section a detailed description of the perception pipeline of the DTI Robot co-worker, that includes 2D, 2.5D and 3D single camera pose estimation is presented. This section will contain a description of the first vision system implemented in the DTI robot co-worker platform that is presented in **Contribution B** and **Contribution C** in the contribution section, Section 2.9. A general description of the technical and non-technical aspects of the DTI Robot co-worker platform are found in the Phd. thesis of Andersen R.H [And15]

2.8.1 Flexible single camera pose estimation

A short overview of the cases in Contribution B and C are presented such that the reader is familiar with the task before going into detail with the vision system. In the two papers following this section, two different cases from two different companies are solved using the DTI Robot Co Worker platform. The cases are not previous automated with conventional robot technology due to the flexibility required in such solutions. What the companies need are flexible and re-configurable solutions. The first cases is a simple Pick n Place task where small aluminium needles have to be moved from one tray to another. It is not automated because the company only has a couple of employees to do the task once a week. The task flow is illustrated in Figure 2.17 (right). The other case in 2.17 (left) is an assembly task where an electronic product must be assembled by picking a transformer from a box and mount it in a heat sink. The second case requires the vision system to compute an accurate 6D pose in order to mount the transformer in the heat sink.

Transformer case:
The detection of the transformer is a classic 2D robot vision application where one "feature" is detected with a pattern recognition algorithm. Pattern training and detection of known pattern(s) is performed by using Mvtec's halcon image processing library. The method is searching a region in an image having most similar shapes to that of trained pattern. The maximum and minimum allowed rotation angles, scale factors, and translations, as well as similarity scores to the trained pattern are parameterized. In the training phase a binary model image is created from an image of the transformer without any other objects in the image, see Figure 2.18 (left, middle). The detection result is shown in Figure 2.18 (right). A object reference frame is calibrated by putting a calibration



(a) Trafo case (b) Needle case

Figure 2.17: The needle and trafo case presented in Contribution B & C

target on top of the empty transformer. Hence, a calibration plane for the 2D application is created, which ensures correct pose computation.

Figure 2.18: Left: The Image model. Middle: The binary model. Right: The detected model

Unlike the transformer detection, the heat sink detection process is divided into 2 stages; global and local feature detection. The reason having additional local feature detection is to estimate the 6D pose of the object to ensure correct assembly of the product. The procedure detecting global feature is largely identical to the transformer detection case; extracting unique feature by setting a specific region of interest in the object. However, because the object is having 6 DoF transform, the global feature detection has to be made with trained pattern scaled in both axis and perspective distortion.

The local feature detection has to locate at least four stable feature points in the image in order to estimate the 6D pose using a Perspective-n-Point algorithm as

described in Section 2.6.2. The local features of the heat sink object are holes. The feature detection algorithm is trained by giving the radius of each hole and its approximate maximum allowed deformation (from perspective distortion). Then synthetic binary images of circles, which fit the training parameters are created. Using this approach is must faster than having multiple circular trained pattern. The location accuracy of the heat sink object is proportional to detection accuracy of local feature points, and detection of a small local features with high precision comes with a processing time when it is asked to search the whole image.

In order to overcome this drawback, an adaptive local search region is introduced. In principal, the search regions are scaled, rotated, and translated based on global feature detection result. As shown in Figure 2.19, the circular search regions in white are adjusted based on the 2D transform result of the global feature of the heat sink object. Search region size for each local feature is set so that it is small enough to make the local feature detection process efficient, but large enough to allow global feature transform estimation error and translational deviation coming from height inequality of global pattern to local features.

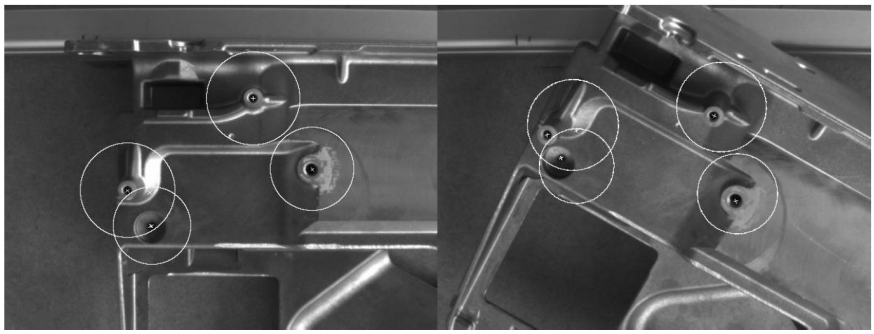


Figure 2.19: Detection of local hole feature points of heat sink object to estimate the 6D pose.

Needle case:
The needle object detection process is similar to the local circular feature detection of heat sink object but without the adaptive search region functionality. A whole image is used when searching object(s) while the deformation parameter is disabled in the needle detection. Therefore, the detection process behaves as if it finds the center of circle(s) in the image, which in this case is performed efficiently. There is an additional feature point to be extracted, which allows

for a 2.5D pose estimation. This will estimate a 4 DoF transform of the object; (Rz, Tx, Ty, Tz). The second feature point having known distance with respect to the center of the circle, is assigned to a point on the rim of each needle detected. Note that the orientation Rz of the needle is not important when picking procedure is conducted, therefore any point on the rim can be used and it is safe to set $Rz = 0$. In figure 2.20, two examples of the 2.5D needle detection are illustrated.

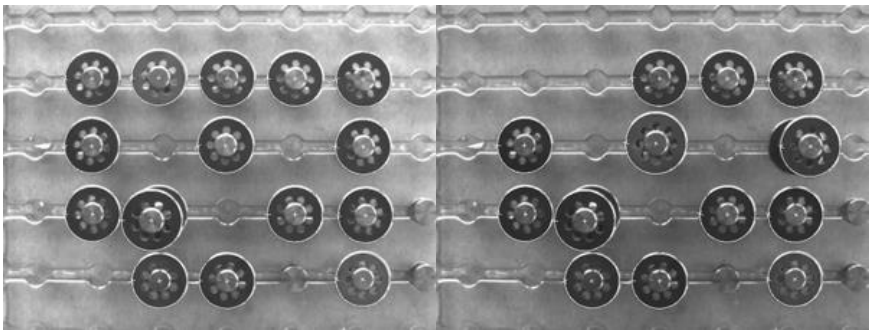


Figure 2.20: Needle detection with 2.5D pose estimation techniques. 4 DoF (Rx, Tx, Ty, Tz) transform of the object is estimated from 2 feature points of known geometry.

2.8.2 Graphical programming of vision tasks

The DTI Robot Co-worker platform features a HTML5 graphical user interface for instructing the task flow. Conceptually, the entire task flow is conducted by building a graph of primitives and skills. A primitive is the lowest level of the control and includes action like "move robot", "close gripper" and "find object". A skill is a composition of primitives, which together create higher level functionality. For more details, see Contribution B and C.

The vision system is embedded in this system by letting a vision job be a low level action equally to a move primitive. In this way the operator can select a locate object primitive on the GUI by dragging the primitive into the list of skills/primitive that represent a given task; just like any other primitives. A sequence of skills and primitives at the GUI is shown in Figure 2.21.

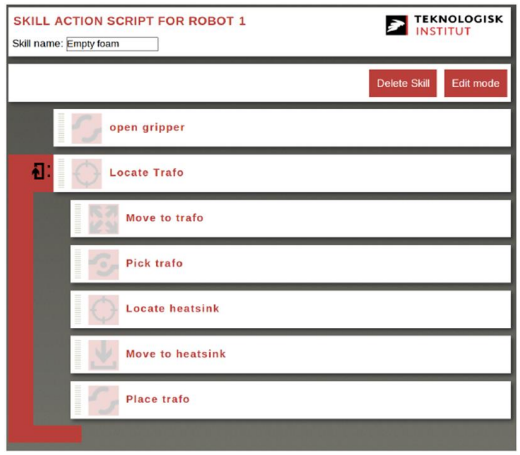


Figure 2.21: Locate Trafo vision primitive in the task flow. An vision primitive are parameterized in the same way as a move primitive

When the skill executor reach the locate object primitive in the execution flow, the vision system is called by the Skill executor in the same way as the robot is "called" when a move primitive is executed at the robot. Like a move primitive, the Locate object primitive has to be parametrized. The parametrization is a matter of training a object model. The user simply trains the 2D pattern(s) required to detect the object(s) by creating a template image. The template image can be a 2D image of the object with no disturbing texture or other objects in the image, a binary image or a synthetic model. A synthetic model is generic model like circles and rectangles. Synthetic models are in the system by default. These types of models are easy to use due to the small amount of parameter associated to the models. Typically, only radius and perspective deformation of the circles model and the length of the sides on the rectangle are needed. Synthetic circle models are used to detect the objects in Figure 2.20. Examples of a real, a binary template model and the detection of the model is shown in Figure 2.18 and the same models are shown in the training GUI in Figure 2.22.

During pattern training some parameters have to be adjusted in the teaching process. Each parameter adjustment can easily be verified by pressing the detect button, see Figure 2.22. Then the vision system will try to detect the model in the live streamed image. In this way the production worker gets instant feedback

Figure 2.22: Graphical user interface for training new objects.

in the parametrization phase and can visually inspect if the adjustment is correct by checking whether the object contour is drawn correctly as shown in Figure 2.22.

The vision system has different pattern training method implemented. Each method is suited for different kind of objects. Method for textured objects uses keypoint detection to compute the position of the object in the image plane. All methods can be selected in the drop down menu named methods in Figure 2.22. When a detection model is created and trained, the model can be selected in a Locate object primitive. The separation of models and primitives makes it possible to reuse detection models in different primitives only with different model runtime parameters and a different reference frame. The reference frame is configured in the primitive by the procedure described in Section 2.3.1.

Figure 2.23 shows the GUI for parameterization of Locate object primitive. The model that is previously trained and configured is selected from a drop down menu. All available models in the system will be visible in the drop down menu. The user simply selects the correct model and configures the local and relative transforms needed to compute the grasp point.

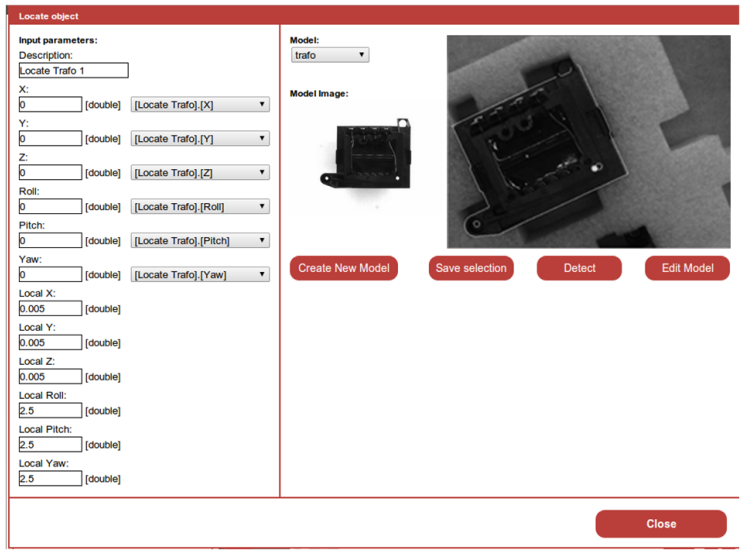


Figure 2.23: Locate object primitive

2.9 Contributions

This section presents the contribution B and C.

Contribution B: *Definition and Initial Case-Based Evaluation of Hardware-Independent Robot Skills for Industrial Robotic Co-Workers*

[Contribution B], entitles "*Definition and Initial Case-Based Evaluation of Hardware-Independent Robot Skills for Industrial Robotic Co-Workers*" is published on the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Workshop on Robotic Assistance Technologies in Industrial Settings held from 3rd-8th of November 2013 in Tokyo, Japan.
The paper presents the studies framework of the DTI Robot Co-worker and discusses the background, the Robotic Skill structure and a few cases.

Contribution C: *Definition of Hardware-Independent Robot Skills for Industrial Robotic Co-workers*

[Contribution C], entitles "*Definition of Hardware-Independent Robot Skills for Industrial Robotic Co-workers*" is published on the 45th international Symposium on Robotic (ISR/Robotic 2014) held from the 2nd to 4th of June 2014 at the Münich Trade Fair Centre, Germany.
The paper presents the framework of the DTI Robot Co-worker and discusses the background, the Robotic Skill structure and a few cases.

Definition of Hardware-Independent Robot Skills for Industrial Robotic Co-workers*

Rasmus Hasle Andersen[1], Thomas Sølund[2] and John Hallam[3]

***Abstract*— In this paper we present a framework which facilitates easy and intuitive robot instruction, allowing non-experts to instruct and use industrial robots.**

The framework is based on flexible, generic and hardware-independent robot Skills based on predefined symbolic unit actions called Primitives. We demonstrate the feasibility of our approach through case studies of real industry tasks which are not automated today, because they would be too expensive given the high cost of (re-)configuration using current automation approaches.

I. INTRODUCTION

(Re-)programming a robot currently requires expert knowledge of robotics as well as process knowledge of the task at hand, and is a cumbersome process even for specialists. This limits the use of robotics in small and medium enterprises (SMEs) with flexible and varying production, since reconfiguration of a robotic installation is too resource-intensive for the installation to be viable. To strengthen the competitiveness of SMEs, the creation of agile robots which can easily be reconfigured for new tasks and operated by existing personnel is needed. We wish to shift focus from robot programming to robot instruction so that non-experts are able to interact directly with an industrial robotic installation. At the Danish Technological Institute (DTI) we investigate the use of hardware-independent robot Skills to facilitate easy instruction by users with no previous robotics experience. Our goal is to produce a robotic co-worker which can be operated by existing shop-floor personnel after a short training course. This paper presents our first steps towards the realization of such a system. At DTI we call our robotic co-worker platform the *DTI Robot CoWorker*. This platform is presented in Fig. 1.

The main challenge is to transfer knowledge sensibly from the human operator to the system. Many have tried to simplify this transition by applying the concept of imitation [1]. Imitation learning is also known as Programming-by-Demonstration (PbD) or Learning-from-Demonstration (LfD). Imitation consists of four phases, each trying to answer one of the following questions:

1) What to imitate?
2) How to imitate?
3) Who to imitate?
4) When to imitate?

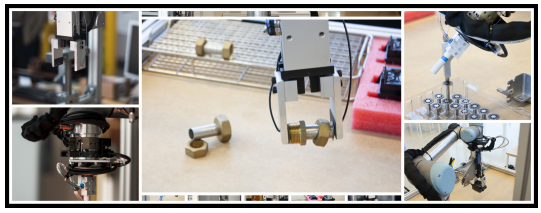


Fig. 1. The DTI Robot CoWorker setup

We focus on the first two questions, i.e. what and how to imitate, by the use of hardware-independent robot Skills and Tasks, such that several hardware configurations can use the same Skill descriptions to complete a given Task. Adopting the arguments from psychology that human movements are possibly built from motor primitives or action units [2], analogous to speech being a composition of phonemes, actions can be built from smaller action units. Additionally actions can themselves have different sizes, thereby representing different levels of abstraction. This philosophy is transferred to and applied within robotics by devising a framework consisting of low-level unit actions called *Primitives*, a structured combination of such Primitives called *Skills* and finally a high level process description called *Tasks*. This layered framework is illustrated in Fig. 2 and serves as the foundation for specifying industrial processes to be solved by our DTI Robot CoWorker.

As indicated in Fig. 2, the operator will only interact with the system from the Skill layer and up; the Primitive layer is meant to be handled by experts. The layered framework serves as a way of abstracting the complexity at the lowest level of execution so that the user can focus on *what* to do, namely the process, by creating a Task description. The expert implementing the Primitives provides the system with knowledge of *how* to perform a specific unit-action. By representing the robot's capabilities in terms of symbolic Primitives, the operator does not need to have specific robot knowledge since an intuitive representation of what the system can do is provided. This gives the operator a clear understanding of what the system can actually do and

*The research leading to these results has been funded in part by the Danish Ministry of Science, Innovation and Higher Education under grant agreement #11-117525 and from the the European Unions seventh framework program (FP7/2007-2013) under grant agreements #285380 (PRACE: The Productive Robot Apprentice) and #287787 (SMErobotics: The European Robotics Initiative for Strengthening the Competitiveness of SMEs in Manufacturing by integrating aspects of cognitive systems).

[1]Rasmus Hasle Andersen is with the Danish Technological Institute, Robot Technology, 5230 Odense M, Denmark raha@dti.dk

[2]Thomas Sølund is with the Danish Technological Institute, Robot Technology, 5230 Odense M, Denmark thso@dti.dk

[3]John Hallam is with the The Maersk Mc-Kinney Moller Institute, University of Southern Denmark, 5230 Odense M, Denmark john@mmmi.sdu.dk

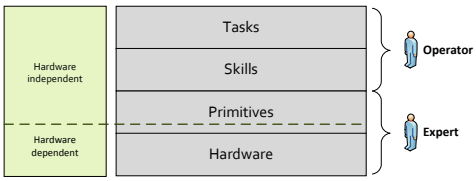


Fig. 2. Hierarchical framework illustrating the different layers. The system operator only uses the Skill and Task layer such that the complexity of the Primitive layer and the hardware itself is and abstracted. Skills and Tasks are hardware-independent, while Primitives are hardware-dependent and implemented by experts.

consequently he has an easier time instructing the system.

The Primitives are configured and specialized for each scenario by adjusting parameters. These parameters can be updated at run-time, hence we enable automatic adaptation to minor changes in the environment.

This paper is structured as follows: Section II contains related work; section III introduces our robotic co-worker system, the DTI Robot co-worker, which serves as the test-platform; while section IV describes our approach to realize hardware-independent robot Skills. Section V describes instruction and execution of Skills and cases studies are presented in section VI.

II. RELATED WORK

As argued in [3], two conceptually different approaches exist when talking about programming by demonstration. One focusses on encapsulating symbolic task knowledge as a function of motion such as the object-action complexes of [4] and the skill primitives in [5]. Another approach focuses on symbolic representation of system knowledge [3], [6], [7]. The idea of modeling continuous actions and observations as Primitives is appealing since this provides a means of dealing with a symbolic representation such that the continuous world is discretized into meaningful symbolic units. Automatic definition of Primitives has proven very difficult [8] and often the Primitives are defined by hand [6]. Our work belongs to this second-symbol based approach.

In [9] a hybrid discrete-continuous supervisory control architecture is applied which activates the correct Primitives (in accordance with a specified task/goal) based on multi-sensor observations. Their architecture allows for combined discrete and continuous control: discrete control while selecting the appropriate Primitive, and continuous control while executing a given Primitive.

Extensive research exists within the field of motor Primitives, but far less exists regarding sensing Primitives, not to mention the combination of the different Primitives [10]–[12]. Mason's original work [12] on the Task Frame Formalism provided the basis for later research to investigate the use of relative frames for task description. Though this approach is still applied in [10] it does not provide a means for easy instruction to the end-user. It still requires an expert to program the robot.

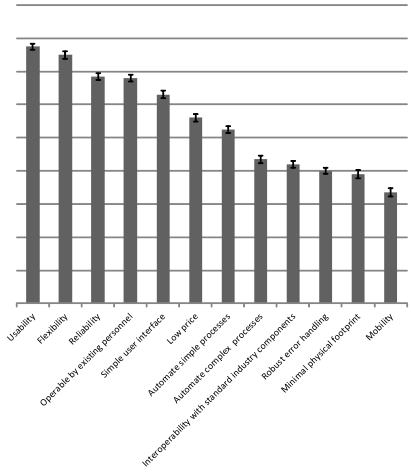


Fig. 3. Relative priority chart of desired features of an industrial robot for SMEs in Denmark. Reproduced with authorization from original author.

Many different approaches have been proposed for implementing Primitives. One branch is based on the pioneering work of Mason's Task Frame Formalism [12]. This branch includes [13]–[16]. Others have proposed encoding Primitives as Hidden Markov Models (HMMs). Billard et al. use HMMs to synthesize trajectories of the robot, and specifically use one HMM per joint in [1], while Calinon et al. use one additional HMM for the end effector in [17].

In [9], [18], [19] a similar hierarchical approach is taken, though they take a bottom-up approach focusing solely on the technological challenges and not on the end user's interaction with the system. In [20] human-robot interaction is tackled using human gestures combined with a Skill-based approach. The European project RoboEarth [11] uses a similar approach to the one proposed here, dividing actions into two categories: system-dependent and system-independent, and facilitates sharing of system-independent Skills between different robots, though within the area of assistive tasks rather than industry-related operations with their appertaining precision requirements.

An important factor in successfully applying robot Skills is sufficient task coverage by the set of Skills available. Bøgh et al. [19] identified a set of 13 Skills sufficient for necessary tasks on the shop-floor at Grundfos A/S, though they did not split the Skills into Primitives as we propose here.

III. SKILLS FOR INDUSTRIAL ROBOTIC CO-WORKERS

There is a huge unexploited potential for using industrial robots in SMEs. Many SMEs request robotic solutions which are flexible, reliable and usable by non-experts. The DTI Robot Co-Worker is a modularized robotic installation which seeks to meet these demands. Its flexibility makes it easily configurable to handle a variety of industrial processes.

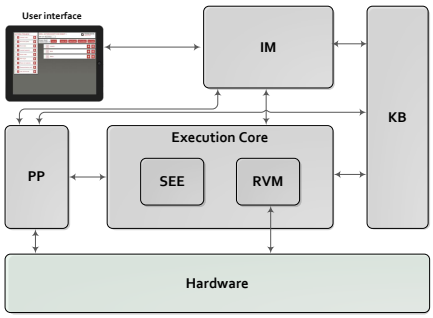


Fig. 4. Simplified system overview presenting the current system. The user instructs the system through the Interaction Manager (IM), and all semantic data in the system is stored in the Knowledge Base (KB). The execution is handled in the execution core by the Skill Execution Engine (SEE) and the Robot Virtual Machine (RVM). Sensor processing is managed in the Perception Pipeline (PP) where semantic information is extracted for use by the Execution Core.

Results from a workshop held by the Danish Industrial Robot Association (DIRA) [21] conclude that Industry requests future robot systems to be user-friendly, flexible, reliable, operable by existing personnel while at the same time securing a return of investment within about 2 years. The attendees were asked to prioritize a list of possible features of a Robot Co-worker; the resulting relative priority chart is shown in Fig. 3. The workshop also investigated which tasks such a system would be expected to solve. 56% of the participants chose "simple picking", "placement in fixture", "simple assembly" and "machine feeding" as the most relevant tasks.

This is a clear indication that a robotic co-worker should be easy and intuitive to use even for users with no previous knowledge of robotics, facilitate flexible production by having focus on automating simple processes, with minimal configuration while maximizing the reusability of previous configurations.

A. System overview

A simplified system overview of our current system is visualized in Fig. 4. The Interaction Manager (IM) is the component facilitating the interaction between the system and the user. Currently interaction using a GUI and a kinesthetic interface are the options available. The Knowledge Base (KB) is the central information storage and interpretation module. All information generated within the system is stored in the central KB, which handles the translation between raw-data and semantic data, which ensures that all modules have a common interpretation of the stored data.

The Execution Core consists of the Skill Execution Engine (SEE) and the Robot Virtual Machine (RVM). Within the RVM the specific interface to the actual hardware is implemented such that hardware interfaces are abstracted and made transparent for the rest of the system. The RVM has direct access to raw sensor data for applications with real-time requirements. The Primitives are directly implemented in the RVM by a system engineer, ensuring correct operation at the lowest level of execution. Hence for a Primitive to be available on a given hardware an implementation specific for that hardware is required. It is important to stress that the RVM only needs modification when new, currently unsupported, hardware is added to the system. In the SEE, high-level Skill execution is handled by analysing the Skill structure and configuring the RVM accordingly, while providing the necessary information from the Perception Pipeline (see below for more information on the Perception Pipeline) and KB. The SEE adjusts the runtime configuration of the RVM and the Primitives to be executed depending on the current execution flow, hereby providing runtime adaptation.

The Perception Pipeline (PP) interprets and fuses raw sensor data to produce semantic data. The PP creates an abstraction of all data processing and sensor fusion algorithms, such that the user of the system only needs to train and parametrize perception models to fit a given Task. Hereby complicated sensor configuration is hidden from the user. At the moment only vision data processing is supported but in the future other sensor modalities will be included, e.g. force/torque sensors to support force-controlled robot motions.

B. Hardware platforms

We are currently testing the concept on two different hardware platforms. One platform consists of a Universal Robot arm (UR5) mounted in a cell with a dual-Asus Xtion-PRO scene camera setup, a Basler ace VGA tool camera, a Stäubli MPS 32 Tool-changer, a suction gripper and a parallel gripper (both grippers are compatible with the tool-changer). The second platform features a COMAU Smart 5 Arch4 arm, a HybridGripper [22] and a Basler ace VGA tool camera.

IV. HARDWARE INDEPENDENT ROBOT SKILLS

One of the reasons for organizing robot action into Skills originates from studies of the human cognitive system [2]. These studies indicate that human cognitive abilities are composed of cognitive primitives (Behaviour Units), and when a specific set/combination of these low-level cognitive primitives is executed, humans are able to understand and extract semantics by identifying these Behaviour Units and their interconnections. Thereby humans are able to cope with very complicated tasks [2], [23]. We apply a similar abstraction in the context of industrial robotic co-workers by using a modularized hierarchical representation of robot capabilities, namely:

1) Primitives
2) Skills
3) Tasks

Primitives serve as the basic building blocks for creating Skills. A Primitive is realized through a hardware-independent module which provides a formal description of the Primitive while an implementation of the Primitive provides the actual functionality. Primitives are divided into two categories, *Motor-Primitives* and *Sensor-Primitives*. Motor-Primitives control actuators such as manipulators and

grippers whereas Sensor-Primitives are high-level interfaces to sensors, providing functionality such as object detection and localization. We introduce Sensor-Primitives to make the sensor system hardware-independent as well. Conceptually this approach gives the opportunity of implementing sensor algorithms on low level controllers, PLCs or powerful computers with hardware acceleration.

Skills are created as structured combinations (graphs) of Primitives and/or other Skills. The logical connections between embedded Skills and Primitives as well as the data-flow between them is specified. Data is shared between Skills and Primitives through the use of parameters. Therefore Skills describe the execution flow.

A *Task* is a definition of what should be accomplished and is described by using a set of goals, as opposed to a Skill which is a description of the execution. Currently we are testing the Primitive and Skill layers, postponing the Task layer until the lower layers have been validated.

Primitives and Skills are the core components of the system. They represent capabilities of the system and are both specific types of action. Primitives can be compared to basic human capabilities such as controlling motion of individual body parts and recognizing objects, whereas Skills are comparable to how we combine motor coordination and object recognition to solve complex problems. In the following we use the term *action* as a generic term to denote either a Primitive or a Skill. Given that a Skill is a composition of other actions, a Skill-hierarchy is created. We distinguish the different levels in this hierarchy by subscripts indicating the level, where $Action_1$ indicates a Skill only containing Primitives, and $Action_0$ represents a Primitive. An example is given in (1) where $Skill_n$ is a Skill from layer n containing the listed actions.

$$Skill_n = (Action_{n-1}, Action_i, Action_0) \tag{1}$$

where $n - 1 \geq i \geq 0$.

The formal definition of Primitives and Skills are given in (2) and (3). A Primitive is defined by a set of input parameters used for detailing the behaviour and a set of output parameters used to share and reuse runtime generated information. Skills are defined as a directed graph, and therefore a Skill has nodes A (actions) and edges C (connections) as additional parameters. The set of nodes in A also contains two control nodes, indicating start and end of the Skill structure.

$$Primitive :=< \mathbf{par_{input}}, \mathbf{par_{output}} > \tag{2}$$
$$Skill :=< \mathbf{A}, \mathbf{C}, \mathbf{par_{input}}, \mathbf{par_{output}} > \tag{3}$$

A parameter is defined in (4) as a triplet consisting of a type, a name and a value.

$$Parameter := < type, name, value > \tag{4}$$

Each action a in $\mathbf{A}$ is either a $Skill$ or $Primitive$, and each connection c in $\mathbf{C}$ is either a data connection (DC) or a flow connection (FC) as defined in (5) and (6).

$$a := \{Skill, Primitive\} \tag{5}$$
$$c := \{DC, FC\} \tag{6}$$

Data connections are used to specify which parameters (if any) are passed between actions. Flow connections specify which action is activated next based on the evaluation of a condition, hence represent the logical flow within a Skill. Both connection types have a source and destination field; DCs have an additional field: the parameter passed from one action to an other, see (7). If multiple parameters are exchanged between actions then one DC is present per parameter. FCs have a $condition$ as the additional field, see (8).

$$DC :=< src, dst, parameter > \tag{7}$$
$$FC :=< src, dst, condition > \tag{8}$$

By using flow connections the behaviour of the Skill is adapted according to the perceived environment and execution status. The $condition$ in a FC is defined by the triplet in (9),

$$cond :=< parameter, operator, value > \tag{9}$$

where $operator$ is of the set

$$operator := \{<, >, \leq, \geq, \neq, =\}$$

A graphical representation of a generic Pick And Place Skill is given in Fig. 5.

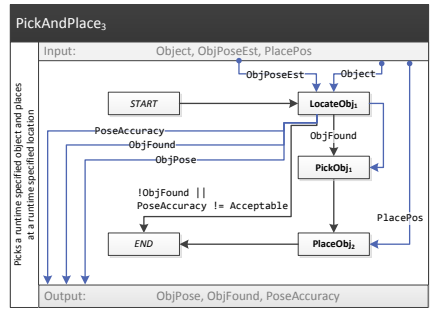


Fig. 5. Graphical representation of a Pick and Place Skill (a level 2 Skill). The internal boxes are actions; bold names indicates Skills and those in italic are special control Primitives. Blue arrows indicate data connections and gray arrows represents flow connections. The input parameters are: *Object*, *ObjectPoseEst* and *PlacePose*. They represent, respectively, the model describing the object to pick, a rough estimate of the object position to ensure the object is in the tool-cameras field of view and a position specifying where to place the object.

V. Instruction and Execution of Skills

Given that we are currently not focussing on Tasks, system interaction concentrates on specifying how to act as opposed to what to achieve. Therefore interaction leads to the specification of a Skill, which is stored in the KB such that other modules can reuse this information at different times as needed.

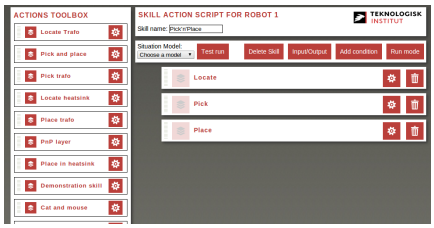


Fig. 6. The graphical user interface for creating new Skill models.

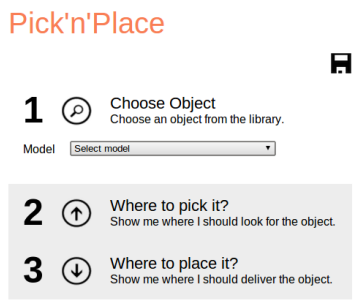


Fig. 7. The wizard based graphical user interface used for instantiating Skill models.

Creating new Skills is a simple process, consisting of connecting existing actions in the desired structure and specifying the external interface in terms of which parameters are required as input and which are produced as output. The simplicity of creating new Skills is indeed one of the main strengths of our system: is does not require robot experts to create new Skills, so even non-experts can easily instruct a robot.

The creation of new Skills is done using the Interaction Manager. Initially a Skill model is created which defines the data flow, execution flow and the external interface in terms of input and output parameters. Having defined a model it needs to be instantiated such that the required input parameters are provided when and where needed. A video presenting an early prototype of the instruction and execution process is available in [24]. The interface used to create Skill models is presented in Fig. 6. Skill instantiation is possible using either the aforementioned interface or using a wizard-based approach visualized in Fig. 7. The wizard-based approach is meant for the shop floor operator, where the operator should just follow the on-screen guidelines to instantiate a Skill and thereby instruct the robot how to act.

When configuring a robot system, it is usually very difficult to configure the sensor processing (especially vision-related) due to the number of parameters typically related to this. The use of Sensor-Primitives and Skills provides a simple approach to perform sensor configuration, since the operator only needs to specify which object to process and detailed sensor configuration is abstracted away. We only handle known objects in our current setup, and each object has a unique id. Information on how a given object is handle by a specific tool is specified in the KB. This allows the system to validate whether a given Skill instance is executable by the current hardware configuration. If the Skill is not executable the IM asks the user to either A) change the tool to one which actually can handle the specified object, or B) teach the system how the object should be handled by the current tool. To ease the process of teaching a new object model to the system we have embedded the Halcon [25] image processing library and created a graphical user interface to train models and store them in the KB. When parts are taught to the system, a Sensor-Primitive is configured simply by selecting the desired model of the part.

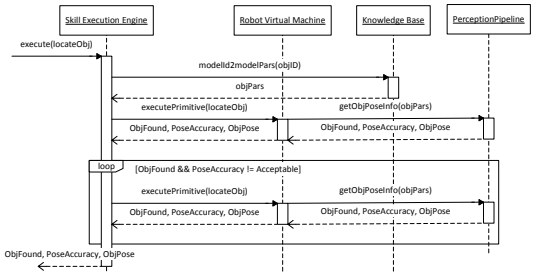


Fig. 8. Sequence diagram of Skill execution showing the interaction between the Skill Execution Engine (SEE), the Robot Virtual Machine (RVM), the Perception Pipeline (PP) and the Knowledge Base (KB) during execution of the Skill $locateObj_1$

A. Skill Execution

The SEE is where the actual execution of hardware-independent Skills is effected. Since focus is purely on Skills and Primitives, and not the low level execution thereof, many of the low-level complexities of execution are abstracted away. When executing a Skill, the SEE analyses the entire Skill structure and handles the execution of and transition between actions (Skills or Primitives). The SEE thereby transforms the hardware-independent Skill into a executable Primitive sequence, and while making this transformation the SEE ensures that the required Primitives are in fact available on the current hardware configuration.

The nominal execution flow of the Skill $LocateObj_1$ is shown in Fig. 8. The SEE retrieves the detailed object parameters from the KB, hence performs an "object id to object parameters" translation the results of which are then passed along to the RVM. When the RVM executes a Sensor-Primitive (such as $LocateObj_0$) the RVM retrieves the actual pose from the PP using the true object parameters previously retrieved by the SEE. After evaluating the accuracy of the estimated pose, the SEE determines whether to optimize the pose depending on the specification of Skill.

As exemplified in Fig. 8 the execution in our current system is sequential, and we currently do not support concurrent execution. This is a topic which we will investigate in the near future. The presented definition of Skills and Primitives does not conflict with parallel execution of Skills and/or Primitives, though the current representation needs to be extended to support it.

VI. Case Studies

To evaluate our hierarchical Skills we have chosen two different, though similar, tasks. Both are pick-and-place operations, but they handle different objects in different contexts and require different hardware (a different tool per object). The two chosen tasks are both found in industry, and we refer to them as Task A and Task B in the following. Task A is a sub-process from an assembly line at a Danish company and Task B is a specific process from a German company. Both tasks are solved manually today, since their automation is too expensive because of the high cost of reconfiguration.

We demonstrate that the same Skill-model can be instantiated to solve both Tasks, hence demonstrating the hardware-independence and flexibility of our proposed hierarchical framework. We have successfully deployed the system on two distinct hardware platforms as mentioned in section III. This allows us to test and create Skills on one platform and transfer them to the other, thereby testing the hardware independence. This extends to any platform for which the required primitives are available. The reported cases have only been tested on the UR-platform due to hardware availability at the time of testing.

We have created a generic *PickAndPlace* Skill, which consist solely of the set in (10),

$$PnP_3 = \{LocateObj_1, PickObj_1, PlaceObj_2\} \qquad (10)$$

namely a locate, pick and a place Skill. The structural configuration is presented in Fig. 5. The Skill has three input parameters: Object (the object to handle), ObjPoseEst (an estimate of where the object is), and PlacePos (the position to place the object). The object is specified through an object id, the estimated object pose is specified through a 6D-pose and the position of the place operation is specified through a generic position parameter. This generic parameter can be either a 6D-pose or a object id. If it is an object id, the $PlaceObj_2$ will internally activate a $LocateObj_1$ skill with the corresponding object id, thereby specifying the position of the place operation through the detection of an object. Consequently simple assembly can also be achieved using this generic Pick and Place Skill. The output is specified by three parameters: ObjPose (the actual 6D-pose of the object), ObjFound (boolean value indicating whether the object was located or not) and PoseAccuracy (value indicating the accuracy of the estimated pose).

Task A consists of picking up a transformer from a known region of interest, and placing it in a heat sink which is in an unknown position. The position of the heat sink and the transformer varies each time, hence reactive execution is required. Task A is visualized in Fig. 9, and consists of four steps: 1) detecting the transformer to be grasped, 2) picking up the transformer, 3) detecting the heat sink in which to place the transformer and 4) placing the transformer in the detected heat sink. The placement location is specified by the object id of the heat sink. This ensures that the placing of the transformer occurs in the heat sink as desired.

Fig. 9. The process of Task A, specified in four different steps, namely detection, picking, detection of place-location and actual placing.

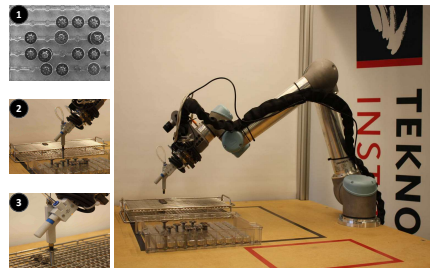


Fig. 10. The process of Task B, specified in three steps. Detection of a needle, picking a needle and placing a needle.

Task B is a process where multiple needles have to be moved from a plastic tray to a metal tray. The task is visualised in Fig. 10), and consists of three steps: 1) detection of the (next) needle to pick up, 2) picking up the needle and 3) placing the needle. The place positions are hand coded in this task, as opposed to the generic place position used in Task A.

VII. Conclusions

We have presented a hardware-independent Skill model, which makes use of a Robot Virtual Machine to control physical hardware. By introducing Skills and Primitives the system can be instructed simply by selecting the desired actions and parametrising them accordingly to the Task at hand. This greatly increases usability and decreases the requirements on the operator in order to use the system. An operator can therefore instruct the system given no previous knowledge of programming or robotics. We have demonstrated that the use of hardware-independent Skills can be used to solve real industrial tasks, with a minimum of configuration while securing high reusability. This has been achieved by using a generic Pick and Place Skill to solve two different Tasks. The presented cases showed that the Skill

was indeed hardware-independent, and generic in terms of the object to handle.

VIII. FUTURE WORK

In future work we will look into the Task layer of the presented hierarchical framework, and we will investigate parallel execution of Primitives and Skills. This will increase execution speed and will allow the DTI Robot Co-Worker to solve more complex Tasks. It also provides the system a way to control and utilize multiple cooperating robots. A foreseeable challenge by introducing concurrent execution is the extra complexity required to configure the system correctly. We are currently expanding the set of available Primitives and Skills so that the number of Tasks which can be solved by the DTI Robot Co-Worker increases. The use of predefined Primitives enables a symbolic description of the actual robot capabilities and thereby facilitates the use of state-of-the-art machine learning and planning algorithms, which we intend to investigate in future work.

REFERENCES

[1] A. Billard, Y. Epars, S. Calinon, S. Schaal, and G. Cheng, "Discovering optimal imitation strategies," *Robotics and Autonomous Systems*, vol. 47, no. 2-3, pp. 69–77, Jun. 2004.

[2] D. Newtson, "Attribution and the unit of perception of ongoing behavior," *Journal of Personality and Social Psychology*, vol. 28, no. 1, pp. 28–38, 1973.

[3] J. Huckaby and H. H. I. Christensen, "A Taxonomic Framework for Task Modeling and Knowledge Transfer in Manufacturing Robotics," *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pp. 94–101, 2012.

[4] N. Krüger, C. Geib, J. Piater, R. P. Petrick, M. Steedman, F. Wörgötter, A. Ude, T. Asfour, D. Kraft, D. Omrčen, A. Agostini, and R. Dillmann, "ObjectAction Complexes: Grounded abstractions of sensorymotor processes," *Robotics and Autonomous Systems*, vol. 59, no. 10, pp. 740–757, Oct. 2011.

[5] T. Hasegawa, T. Suehiro, and K. Takase, "A model-based manipulation system with skill-based execution in unstructured environment," in *Fifth International Conference on Advanced Robotics 'Robots in Unstructured Environments*, vol. 8, no. 5. IEEE, 1991, pp. 970–975 vol.2.

[6] S. Ekvall, D. Aarno, and D. Kragic, "Task Learning Using Graphical Programming and Human Demonstrations," in *ROMAN 2006 - The 15th IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, Sep. 2006, pp. 398–403.

[7] T. Abbas and B. a. MacDonald, "Generalizing topological task graphs from multiple symbolic demonstrations in programming by demonstration (PbD) processes," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, May 2011, pp. 3816–3821.

[8] V. Krüger, D. Kragic, A. A. Ude, and C. Geib, "The Meaning of Action: a review on action recognition and mapping," *Advanced Robotics*, vol. 21, no. 13, pp. 1473—-1501, 2007.

[9] G. Milighetti, H.-B. Kuntze, C. Frey, B. Diestel-Feddersen, and J. Balzer, "On a primitive skill-based supervisory robot control architecture," in *ICAR '05. Proceedings., 12th International Conference on Advanced Robotics, 2005.* IEEE, 2005, pp. 141–147.

[10] J. D. Schutter, T. D. Laet, J. Rutgeerts, W. Decré, R. Smits, E. Aertbelien, K. Claes, and H. Bruyninckx, "Constraint-based Task Specification and Estimation for Sensor-Based Robot Systems in the Presence of Geometric Uncertainty," *The International Journal of Robotics Research*, vol. 26, no. 5, pp. 433–455, May 2007.

[11] M. Waibel, M. Beetz, J. Civera, R. D'Andrea, J. Elfring, D. Gálvez-López, K. Häussermann, R. Janssen, J. Montiel, A. Perzylo, B. Schieß le, M. Tenorth, O. Zweigle, and R. De Molengraft, "RoboEarth," *IEEE Robotics & Automation Magazine*, vol. 18, no. 2, pp. 69–82, Jun. 2011.

[12] M. T. Mason, "Compliance and Force Control for Computer Controlled Manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, no. 6, pp. 418–432, 1981.

[13] J. Baeten, H. Bruyninckx, and J. De Schutter, "Shared control in hybrid vision/force robotic servoing using the task frame," in *IEEE/RSJ International Conference on Intelligent Robots and System*, vol. 3. IEEE, 2002, pp. 2128–2133.

[14] M. Eng, K. U. Leuven, J. D. Schutter, and H. Bruyninckx, "Where does the Task Frame go?" in *International Symposium of Robotics Research*, Hayama, 1997.

[15] T. Kroger, B. Finkemeyer, U. Thomas, and F. M. Wahl, "Compliant motion programming: The task frame formalism revisited," *Mechatronics and Robotics*, pp. 1029–1034, 2004.

[16] U. Thomas, F. Wahl, J. Maass, and J. Hesselbach, "Towards a new concept of robot programming in high speed assembly applications," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Ieee, 2005, pp. 3827–3833.

[17] S. Calinon, F. Guenter, and A. Billard, "Goal-Directed Imitation in a Humanoid Robot," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, no. April. IEEE, 2005, pp. 299–304.

[18] U. Thomas, G. Hirzinger, B. Rumpe, C. Schulze, and A. Wortmann, "A New Skill Based Robot Programming Language Using UML/P Statecharts," in *Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 461–466.

[19] S. Bøgh, O. S. Nielsen, M. R. Pedersen, V. Krüger, and O. Madsen, "Does your Robot have Skills?" in *Proceedings of the 43rd International Symposium on Robotics*. Taipei, Taiwan: VDE Verlag GMBH, 2012.

[20] M. R. Pedersen, C. Hoilund, and V. Kruger, "Using human gestures and generic skills to instruct a mobile robot arm in a feeder filling scenario," in *2012 IEEE International Conference on Mechatronics and Automation*. IEEE, Aug. 2012, pp. 243–248.

[21] M. T. Nibe, "Opsamling på resultater fra behov og businesscase," Tech. Rep., 2013.

[22] D. T. Institute, "HybridGripper - Flexible gripping," 2013. [Online]. Available: http://www.youtube.com/watch?v=XWR-o-eD_dw

[23] D. Newtson, G. Engquist, and J. Bois, "The objective basis of behavior units," *Journal of Personality and Social Psychology*, vol. 35, no. 12, pp. 847–862, 1977.

[24] R. Hasle Andersen, "DTI Co-Worker: Instruction and Execution (Prototype)," 2012. [Online]. Available: https://dl.dropbox.com/u/19105731/Robot_instruction.mp4

[25] MVTEC, "Halcon Image processing Library," 2013. [Online]. Available: http://www.mvtec.com/halcon/

Definition and Initial Case-Based Evaluation of Hardware-Independent Robot Skills for Industrial Robotic Co-Workers

Rasmus Hasle Andersen, Danish Technological Institute, Robot Technology, raha@dti.dk, Denmark
Thomas Sølund, Danish Technological Institute, Robot Technology, thso@dti.dk, Denmark
John Hallam, University of Southern Denmark, john@mmmi.sdu.dk, Denmark

Abstract

We propose a hierarchical action framework which facilitates easy and intuitive robot instruction, allowing non-experts to instruct and use industrial robots. The framework is based on flexible, generic and hardware-independent robot Skills, which are executed through the use of a Robot Virtual Machine. We demonstrate the feasibility of our approach through case studies of real industrial tasks which are not automated today, due to the high cost of reconfiguration.

1 Introduction

(Re-)programming a robot currently requires expert knowledge of robotics as well as process knowledge of the task at hand, and is a cumbersome process even for specialists. This limits the use of robotics in small and medium enterprises (SMEs) with flexible and varying production, since reconfiguration of a robotic installation is too resource-intensive for the installation to be viable. To strengthen the competitiveness of SMEs, the creation of agile robots which can easily be reconfigured for new tasks and operated by existing personnel is needed. At the Danish Technological Institute (DTI) we investigate the use of hardware-independent robot Skills to facilitate easy instruction by users with no previous robotics experience, and to enable skill reuse across tasks and different hardware platforms. Our goal is to produce a robotic co-worker which can be operated by existing shop-floor personnel after a short training course, by shifting focus from robot programming to robot instruction such that non-experts are able to interact directly with an industrial robotic installation. We propose using a hybrid instruction concept combining process description using a graphical interface and detail specification using kinesthetic instruction and teleoperation. This paper presents our first steps towards the realization of such a system. We refer to our robotic co-worker platform as the *DTI Robot CoWorker*.

Adopting the arguments from psychology that human movements are possibly built from motor primitives or action units [15], analogous to speech being a composition of phonemes, actions can be built from smaller action units. Additionally actions can themselves have different sizes, thereby representing different levels of abstraction. This philosophy is transferred to and applied within robotics by devising a framework consisting of low-level unit actions called *Primitives*, a structured combination of such Primitives called *Skills* and finally a high level process description called *Tasks*. This layered framework serves as the foundation for specifying industrial processes to be solved by our DTI Robot CoWorker.

The operator interacts only with the system from the Skill layer and up; the Primitive layer is meant to be handled by experts. The layered framework serves as a way of abstracting the complexity at the lowest level of execution so that the user can focus on *what* to do, namely the process, by creating a Task description. The expert implementing the Primitives provides the system with knowledge of *how* to perform a specific unit-action. By representing the robot's capabilities in terms of symbolic Primitives, the operator does not need to have specific robot knowledge since an intuitive representation of what the system can do is provided.

2 Related work

As argued in [8], two conceptually different approaches exist to represent robot system knowledge. One focuses on encapsulating symbolic task knowledge as a function of motion such as the object-action complexes in [11] and the skill primitives in [7]. Another approach focuses on symbolic representation of system knowledge [1, 8]. The idea of modelling continuous actions and observations as Primitives is appealing since this provides a means of dealing with a symbolic representation such that the continuous world is discretized into meaningful symbolic units. Our work belongs to this second-symbol based approach. Automatic definition of Primitives has proven very difficult [12] and often the Primitives are defined by hand [6].

In [14,19] a hierarchical approach similar to the one taken here is used, though they take a bottom-up approach focusing on the technological challenges and not on the end user's interaction with the system. In [17] Task instruction is tackled using human gestures combined with a Skill-based approach, though the proposed Skills are not hardware-independent and the instruction does not address the need for detailed process information in manufacturing.The European project RoboEarth [21] divides actions into two categories: system-dependent and system-independent, and facilitates sharing of system-

ISBN 978-3-8007-3601-0

independent Skills between different robots, though within the area of assistive tasks rather than industry-related operations with their appertaining precision requirements.
Extensive research exists within the field of motor Primitives, but far less exists regarding sensing Primitives, not to mention the combination of the different Primitives [18, 21]. Many different approaches have been proposed for implementing Primitives. Mason's original work [13] on the Task Frame Formalism provided the basis for later research to investigate the use of relative frames for task description [2, 4, 10, 20]. Though this approach is still applied in [18] it does not provide a means for easy instruction to the end-user, and it still requires an expert to program the robot.
Others have proposed encoding motor Primitives as Hidden Markov Models (HMMs). In [3] HMMs are used to synthesize trajectories of the robot by applying one HMM per joint, while [5] introduces one additional HMM for the end effector.

3 Skills for Industrial Robotic Co-Workers

There is a substantial unexploited potential for using industrial robots in SMEs. Many SMEs request robotic solutions which are flexible, reliable and usable by non-experts. The DTI Robot CoWorker is a modularized robotic installation which seeks to meet these demands. Its flexibility makes it easily configurable to handle a variety of industrial processes.
Results from a workshop held by the Danish Industrial Robot Association (DIRA) [16] conclude that Industry requests future robot systems to be user-friendly, flexible, reliable, operable by existing personnel while at the same time securing a return of investment within about 2 years. The workshop also investigated which tasks such a system would be expected to solve. 56% of the participants chose "simple picking", "placement in fixture", "simple assembly" and "machine feeding" as the most relevant tasks.
This is a clear indication that a robotic co-worker should be easy and intuitive to use even for users with no previous knowledge of robotics, facilitate flexible production by having focus on automating simple processes, with minimal configuration while maximizing the reusability of previous configurations. We propose to meet these requirements though the DTI Robot CoWorker, based on hardware-independent robot Skills.

3.1 System Overview

A simplified architecture overview of our current system is visualized in **Figure 1**. The Interaction Manager (IM) is the component facilitating the interaction between the system and the user. The Knowledge Base (KB) is the central information storage and interpretation module. All information generated within the system is stored in the central KB, which handles the translation between raw-data and semantic data, which ensures that all modules have a common interpretation of the stored data.

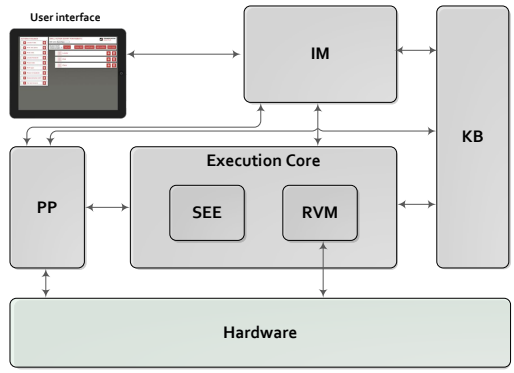


Figure 1: Simplified system overview presenting the current system. The user instructs the system through the Interaction Manager (IM), and all semantic data in the system is stored in the Knowledge Base (KB). The execution is handled in the execution core by the Skill Execution Engine (SEE) and the Robot Virtual Machine (RVM). Sensor processing is managed in the Perception Pipeline (PP) where semantic information is extracted for use by the Execution Core.

The Execution Core consists of the Skill Execution Engine (SEE) and the Robot Virtual Machine (RVM). Within the RVM the specific interface to the actual hardware is implemented such that hardware interfaces are abstracted and made transparent for the rest of the system. The RVM has direct access to raw sensor data for applications with real-time requirements. The Primitives are directly implemented in the RVM by a system engineer, ensuring correct operation at the lowest level of execution. Hence for a Primitive to be available on a given hardware an implementation specific for that hardware is required. The SEE adjusts the runtime configuration of the RVM and the Primitives to be executed depending on the current execution flow, hereby providing runtime adaptation.
The Perception Pipeline (PP) interprets and fuses raw sensor data to produce semantic data. The PP creates an abstraction of all data processing and sensor fusion algorithms, such that the user of the system only needs to train and parametrize perception models to fit a given Task. Hereby complicated sensor configuration is hidden from the user. At the moment only vision data processing is supported but in the future other sensor modalities will be included, e.g. force/torque sensors to support force-controlled robot motions.

3.2 Hardware Platforms

We are currently testing the concept on two different hardware platforms. One platform consists of a Universal Robot arm (UR5) mounted in a cell with a dual-Asus

ISBN 978-3-8007-3601-0

Xtion-PRO scene camera setup, a Basler ace VGA tool camera, a Stäubli MPS 32 Tool-changer, a suction gripper and a parallel gripper (both grippers are compatible with the tool-changer). The second platform features a COMAU Smart 5 Arch4 arm, a HybridGripper [9] and a Basler ace VGA tool camera. The platforms are presented in **Figure 2**.

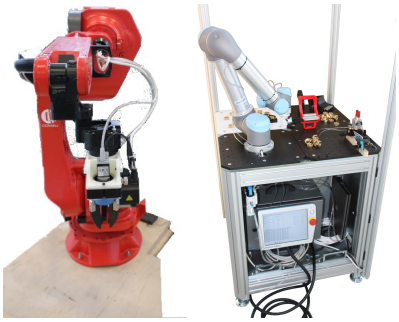


Figure 2: The two hardware platforms; the Comau platform to the left and the UR cell to the right.

4 Hardware-Independent Skills for Industrial Robots

Studies of the human cognitive system indicates that human cognitive abilities are composed of cognitive primitives (Behaviour Units) [15]. We apply a similar abstraction in the context of industrial robotic co-workers by using a modularized hierarchical representation of robot actions, namely: Primitives, Skills and Tasks. This grounds system capabilities in a hierarchy of symbols. Such an approach requires the system to have inherent knowledge regarding the relationships between the symbols and actual execution of such symbols but allows modelling of more high-level processes. The inherent knowledge is provided in the RVM by the expert implementing the primitives.

In **Figure 3** the relation between Actions, Skills and Primitives is presented as a UML class diagram.

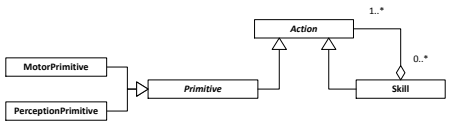


Figure 3: Relationship between Actions, Skills and Primitives as a UML class diagram

Actions serve as an abstract superclass from which *Primitives* and *Skills* are specializations. All *Actions* have input and output parameters which can be used to specialize the behaviour for different circumstances. *Primitives* serve as the basic building blocks for creating Skills. A Primitive is realized through a hardware-independent module which provides a formal description of the Primitive while a Robot Virtual Machine provides the actual functionality. Two types of primitives exists; Motor-Primitives and Perception-Primitives. *Motor-Primitives* control actuators such as manipulators and grippers whereas *Perception-Primitives* are object-centric interfaces to sensors, providing functionality such as object detection and localization.

Skills are defined as directed graphs, the nodes being the Actions, and the connecting edges specifying the interactions and data flow between the action nodes. With the introduction of object-centric Perception-Primitives, object-centric Skills are easily be created such that the dynamics of the environment can be abstracted (to a certain point).

A *Task* is a process description and is described by a set of fully parametrized Skills. The operator will only interact with the system from the Skill layer and up; the Primitive layer is meant to be handled by experts. Thereby the operator can either create new Skills from existing actions (Skills and Primitives), or create Tasks by connecting and parametrizing existing Skills.

Primitives and Skills are the core components of the system. They represent capabilities of the system and are both specific types of action. Primitives can be compared to basic human capabilities such as controlling motion of individual body parts and recognizing objects, whereas Skills are comparable to how we combine motor coordination and object recognition to solve complex problems.

4.1 Definition of Hardware-Independent Robot Skills

Given that a Skill is a composition of actions, a Skill-hierarchy is created. We distinguish the different levels in this hierarchy by subscripts indicating the level, where $Action_1$ indicates a Skill only containing Primitives, and $Action_0$ represents a Primitive.

The formal definition of Primitives and Skills are given in (1) and (2). A Primitive is defined by a set of input parameters used for detailing the behaviour and a set of output parameters used to share and reuse runtime generated information. Skills are defined as a directed graph, and therefore a Skill has nodes A (actions) and edges C (connections) as additional parameters. The set of nodes in A also contains two control nodes, indicating start and end of the Skill structure.

$$Primitive := < \mathbf{par_{input}}, \mathbf{par_{output}} > \quad (1)$$

$$Skill := < \mathbf{A}, \mathbf{C}, \mathbf{par_{input}}, \mathbf{par_{output}} > \quad (2)$$

A parameter is defined in (3) as a triplet consisting of a type, a name and a value.

$$Parameter := < type, name, value > \quad (3)$$

Each action a in $\mathbf{A}$ is either a $Skill$ or $Primitive$, and each connection c in $\mathbf{C}$ is either a data connection (DC)

ISBN 978-3-8007-3601-0

or a flow connection (FC) as defined in (4) and (5).

$$a := \{Skill, Primitive\} \quad (4)$$

$$c := \{DC, FC\} \quad (5)$$

Data connections are used to specify which parameters (if any) are passed between actions. Flow connections specify which action is activated next based on the evaluation of a condition, hence represent the logical flow within a Skill. Both connection types have three fields, where the first two are a source and destination field; the third field of DCs specifies the parameter passed from one action to another, see (6). If multiple parameters are exchanged between actions then one DC is present per parameter. FCs have a $condition$ as the third field, see (7).

$$DC :=< src, dst, parameter > \quad (6)$$

$$FC :=< src, dst, condition > \quad (7)$$

By using flow connections the behaviour of the Skill is adapted according to the perceived environment and execution status. The output from a Perception-Primitives can be used as an input in a Motor-Primitive, or any other action for that matter, to ensure run-time adaptation, such that the system for instance only tries to grasp objects when their presence has been verified. The $condition$ in a FC is defined by the triplet in (8),

$$cond :=< parameter, operator, value > \quad (8)$$

where $operator$ is of the set

$$operator := \{<, >, \leq, \geq, \neq, =\}$$

A graphical representation of a generic Pick And Place Skill is given in **Figure 4**.

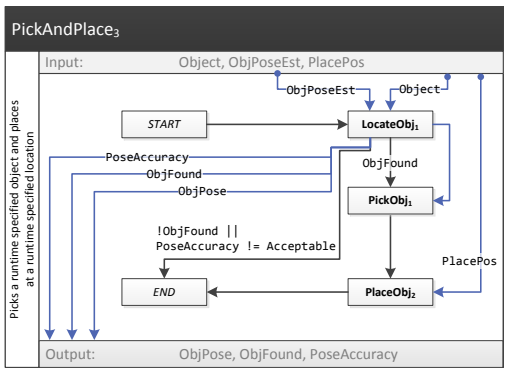


Figure 4: Graphical representation of a Pick and Place Skill (a level 3 Skill). The internal boxes are actions; bold names indicates Skills and those in italic are special control Primitives. Blue arrows indicate data connections and gray arrows represents flow connections. The input parameters are: *Object*, *ObjectPoseEst* and *PlacePose*. They represent, respectively, the model describing the object to pick, a rough estimate of the object position to ensure the object is in the tool-cameras field of view and a position specifying where to place the object.

4.2 Execution of Hardware-Independent Skills

The SEE is where the actual execution of hardware-independent Skills is effected. Skill execution is handled by analysing the Skill structure and configuring the RVM accordingly, while exchanging the necessary information with the Perception Pipeline. A Skill is a modelled system capability which can be reused across different tasks. The reusability of the Skills is ensured by the use of object-centric parameters specifying the actual behaviour of a Skill in a given situation. These parameters can be hardcoded or dynamically provided though runtime parameter exchange between actions. Skills are only executable when they are fully parametrized. The SEE adjusts the runtime configuration of the RVM and adjusts parameters of the Primitives to be executed depending on the current execution flow, hereby providing runtime adaptation. When executing a Skill, the SEE analyses the entire Skill structure and handles the execution of and transition between actions (Skills or Primitives). The SEE thereby transforms the hardware-independent Skill into an executable Primitive sequence, and while making this transformation the SEE ensures that the required Primitives are in fact available on the current hardware configuration.

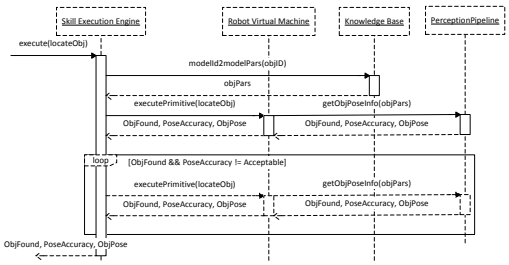


Figure 5: Sequence diagram of Skill execution showing the interaction between the Skill Execution Engine (SEE), the Robot Virtual Machine (RVM), the Perception Pipeline (PP) and the Knowledge Base (KB) during execution of the Skill $locateObj_1$

The nominal execution flow of the Skill $LocateObj_1$ is shown in **Figure 5**. The SEE retrieves the detailed object parameters from the KB, hence performs an "object id to object parameters" translation the results of which are then passed along to the RVM. When the RVM executes a Perception-Primitives (such as $LocateObj_0$) the RVM retrieves the actual pose from the PP using the true object parameters previously retrieved by the SEE. After evaluating the accuracy of the estimated pose, the SEE determines whether to optimize the pose depending on the specification of Skill.

As exemplified in Figure 5 the execution in our current system is sequential, and we currently do not support concurrent execution. This is a topic which we will investigate in the near future. The presented definition of Skills

ISBN 978-3-8007-3601-0

and Primitives does not conflict with parallel execution of Skills and/or Primitives, though the current representation needs to be extended to support it.

5 System Instruction

The process of instructing a robot puts demands on the operator with respect to understanding the actual capabilities of the system and how to apply these capabilities to complete a given task. We encapsulate the complexity of the programming process in simple and intuitive actions, thereby representing the actual capabilities of the robot system. We have created a very simple and intuitive touch based interface for instructing the most typical cases such as pick and place, palletizing and machine tending. The operator is presented with a wizard requesting specific information necessary to detail the process. The operator will be asked to provide details using kinesthetic teaching or teleoperation to minimize the complexity during instruction, thereby exploiting the individual modalities of the hybrid instruction concept.
The result from configuring the process using this wizard-based approach is a Task description consisting of fully parametrized Skills and Primitives. When instructing processes which are not suitable for the wizard based interface, a more advanced interface is available. This interface allows the operator to create new Skills, and to specify a process in more detail, but naturally requires a better system understanding by the operator. Creating new Skills is a simple procedure, consisting of connecting existing actions in the desired structure and specifying which parameters are required as input and which are produced as output. The creation of Tasks is very similar, the differences being that Tasks do not have input or output parameters and a Task consists only of Skills; Primitives are not used in high-level Task descriptions. The simplicity of creating new Skills and Tasks is indeed one of the main strengths of our system: it does not require robot experts to instruct industrial robots.

6 Case Studies

We have evaluated the proposed hierarchical action framework using a set of real industrial manipulation cases which are not automated today due to the high cost of (re-)configuration. We demonstrate (I) the hardware-independence and flexibility of our proposed hierarchical framework by using the same Task description to solve the same process on different hardware platforms in different environments. We demonstrate (II) the reusability across tasks by using the same skill to solve different tasks and finally we demonstrate (III) the simplicity of the system by having non-trained personal create a Task description by instructing the system using our hybrid instruction interface. (I) is evaluated by handling the same process (Task A) on two different hardware platforms using the same Task description. The platforms have previously been described in section 3.2. The process of Task A is visualized in **Figure 6**; the process consists of three steps: 1) detection of the (next) needle to pick, 2) picking up the needle and 3) placing the needle. In the first step dynamic frame referencing is performed to ensure adaptation to run-time variation of the tray position (within a given area). The place positions in the third step are provided as a parameter describing the pattern in the metal tray. This entire process is handled manually today since automation has proven unfeasible. Our two test platforms have successfully solved this process by executing the same Task description, thereby demonstrating the hardware-independence of the concept.

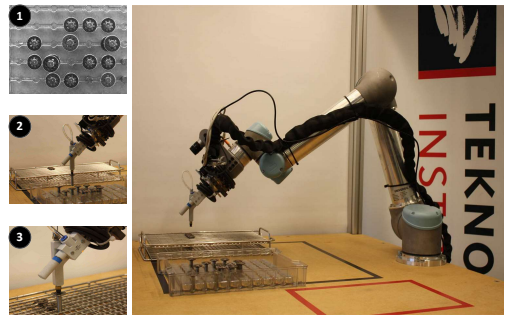


Figure 6: The process of Task A used to validate the hardware independence (I) consisting of three steps. 1) Detection of a needle, 2) picking a needle and 3) placing a needle.

In Task A we solved a process of moving multiple objects from one tray to another. In a different process (Task B) we have reused some of the Skills, thereby demonstrating the reuse of Skills across different tasks (II). An example is the reuse of the *PickAndPlace* Skill. Task B consists of four steps: 1) detecting the transformer to be grasped, 2) picking up the transformer, 3) detecting the heat sink in which to place the transformer and 4) placing the transformer in the detected heat sink. The placement location is specified by the detection of the heat sink. This ensures that the placing of the transformer occurs in the heat sink as desired.

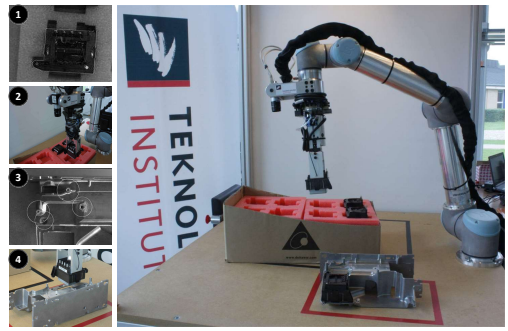


Figure 7: The process of Task B, specified in four different steps, namely 1) detection, 2) picking, 3) detection of place-location and 4) actual placing.

ISBN 978-3-8007-3601-0

The position of the heat sink and the transformer varies each time, hence reactive execution is required. This process is visualized in **Figure 7**.
The structural representation of the object-centric *PickAndPlace* Skill was shown in Figure 4. The Skill consists solely of the set of actions in (9),

$$PnP_3 = \{LocateObj_1, PickObj_1, PlaceObj_2\} \quad (9)$$

The Skill has three input parameters: Object (the object to handle), ObjPoseEst (an estimate of where the object is), and PlacePos (the position to place the object). The object is specified through an object id, the estimated object pose is specified through a 6D-pose and the position of the place operation is specified through a generic position parameter, which can be either a 6D-pose or a object id. If it is an object id, the $PlaceObj_2$ will internally activate a $LocateObj_1$ skill with the corresponding object id, thereby specifying the position of the place operation through the detection of an object. The output is specified by three parameters: ObjPose (the actual 6D-pose of the object), ObjFound (boolean value indicating whether the object was located or not) and PoseAccuracy (value indicating the accuracy of the estimated pose).
We evaluated our hybrid instruction concept (III) by having a non-technical, non-trained person instruct 3 different tasks. The first task is a simple pick and place operation; the second is Task A, the third is Task B, both described above. The user was able to instruct all cases successfully within the first 5 trials emphasising the intuitiveness of the system. The instruction process is an interactive configuration initiated by the operator. The user selects the type of Task, and then the system requests the user to provide the required information through a wizard. The user provides information either by entering/selecting values using the graphical interface, by using kinesthetic teaching, or by teleoperation. The user was told what industrial processes the system should be instructed to perform but had no knowledge of how to solve such processes with robots.

7 Conclusion

By abstracting robot actions as Skills and Primitives the capabilities of the robotic co-worker system are grounded as simple and intuitive symbols, enabling the operator to understand the actual capabilities of the system. Combining this with our hybrid instruction approach, using intuitive touch based graphical interfaces, kinesthetic teaching and teleoperation, we facilitate easy and intuitive robot instruction such that an operator with no previous knowledge of programming or robotics can instruct the system. Through a set of use cases we demonstrate that our approach is applicable to tasks which are not automated today due to the high cost of reconfiguration using current automation approaches. Our approach enables fast and flexible reconfiguration, and thereby increases the usability of robots in small and medium enterprises by decreasing the cost of system reconfiguration. By proving our concept on such different robots as the UR5 and the COMAU Smart 5 Arch4 arm, we showcase the flexibility and conclude that the concept can be applied to all robot types.

8 Future work

In future work we will investigate parallel execution of Primitives and Skills. This will increase execution speed and will allow the DTI Robot CoWorker to solve more complex Tasks. It also provides the system with a way to control and utilize multiple cooperating robots. A foreseeable challenge of introducing concurrent execution is the extra complexity required to configure the system correctly. We are currently expanding the set of available Primitives and Skills so that the number of Tasks which can be solved by the DTI Robot CoWorker increases. The use of predefined Primitives enables a symbolic description of the actual robot capabilities and thereby facilitates the use of state-of-the-art machine learning and planning algorithms, which we intend to investigate in future work.

9 Acknowledgements

The research leading to these results has been funded in part by the Danish Ministry of Science, Innovation and Higher Education under grant agreement #11-117525 and from the the European Union's seventh framework program (FP7/2007-2013) under grant agreements #285380 (PRACE: The Productive Robot Apprentice) and #287787 (SMErobotics: The European Robotics Initiative for Strengthening the Competitiveness of SMEs in Manufacturing by integrating aspects of cognitive systems).

References

[1] Tanveer Abbas and Bruce a. MacDonald, *Generalizing topological task graphs from multiple symbolic demonstrations in programming by demonstration (PbD) processes*, International Conference on Robotics and Automation, IEEE, May 2011, pp. 3816–3821.

[2] Johan Baeten, Herman Bruyninckx, and Joris De Schutter, *Shared control in hybrid vision/force robotic servoing using the task frame*, International Conference on Intelligent Robots and System, vol. 3, IEEE, 2002, pp. 2128–2133.

[3] Aude Billard, Yann Epars, Sylvain Calinon, Stefan Schaal, and Gordon Cheng, *Discovering optimal imitation strategies*, Robotics and Autonomous Systems **47** (2004), no. 2-3, 69–77.

[4] Herman Bruyninckx and Joris De Schutter, *Where does the Task Frame go?*, International Symposium of Robotics Research (Hayama), 1997.

ISBN 978-3-8007-3601-0

[5] Sylvain Calinon, Florent Guenter, and Aude Billard, *Goal-Directed Imitation in a Humanoid Robot*, International Conference on Robotics and Automation, no. April, IEEE, 2005, pp. 299–304.

[6] Staffan Ekvall, Daniel Aarno, and Danica Kragic, *Task Learning Using Graphical Programming and Human Demonstrations*, International Symposium on Robot and Human Interactive Communication, IEEE, September 2006, pp. 398–403.

[7] Tsutomu Hasegawa et al., *A model-based manipulation system with skill-based execution in unstructured environment*, International Conference on Advanced Robotics 'Robots in Unstructured Environments, vol. 8, IEEE, 1991, pp. 970–975.

[8] Jacob Huckaby, Stavros Vassos, and Henrik I. Christensen, *Planning with a Task Modeling Framework in Manufacturing Robotics*, International Conference on Intelligent Robots and Systems, 2013.

[9] Danish Technological Institute, *HybridGripper - Flexible gripping*, 2013.

[10] Torsten Kroger, Bernd Finkemeyer, Ulrike Thomas, and Friedrich M. Wahl, *Compliant motion programming: The task frame formalism revisited*, Mechatronics and Robotics (2004), 1029–1034.

[11] Norbert Krüger, Christopher Geib, Justus Piater, Ronald P.A. Petrick, Mark Steedman, Florentin Wörgötter, Aleš Ude, Tamim Asfour, Dirk Kraft, Damir Omrčen, Alejandro Agostini, and Rüdiger Dillmann, *Object-Action Complexes: Grounded abstractions of sensory-motor processes*, Robotics and Autonomous Systems **59** (2011), no. 10, 740–757.

[12] Volker Krüger, Danica Kragic, Aleš Ude, and Christopher Geib, *The Meaning of Action: a review on action recognition and mapping*, Advanced Robotics **21** (2007), no. 13, 1473–1501.

[13] Matthew Thomas Mason, *Compliance and Force Control for Computer Controlled Manipulators*, Transactions on Systems, Man, and Cybernetics **11** (1981), no. 6, 418–432.

[14] G. Milighetti, H.-B. Kuntze, C.W. Frey, B. Diestel-Feddersen, and J. Balzer, *On a primitive skill-based supervisory robot control architecture*, International Conference on Advanced Robotics, no. 2, IEEE, 2005, pp. 141–147.

[15] Darren Newtson, *Attribution and the unit of perception of ongoing behavior*, Journal of Personality and Social Psychology **28** (1973), no. 1, 28–38.

[16] Malene Tofveson Nibe, *Opsamling på resultater fra behov og businesscase*, Tech. report, 2013.

[17] Mikkel Rath Pedersen, Carsten Hoilund, and Volker Kruger, *Using human gestures and generic skills to instruct a mobile robot arm in a feeder filling scenario*, International Conference on Mechatronics and Automation, IEEE, August 2012, pp. 243–248.

[18] Joris De Schutter, Tinne De Laet, Johan Rutgeerts, Wilm Decré, Ruben Smits, Erwin Aertbelien, Kasper Claes, and Herman Bruyninckx, *Constraint-based Task Specification and Estimation for Sensor-Based Robot Systems in the Presence of Geometric Uncertainty*, The International Journal of Robotics Research **26** (2007), no. 5, 433–455.

[19] Ulrike Thomas, Gerd Hirzinger, Bernhard Rumpe, Christoph Schulze, and Andreas Wortmann, *A New Skill Based Robot Programming Language Using UML/P Statecharts*, International Conference on Robotics and Automation, IEEE, 2013, pp. 461–466.

[20] Ulrike Thomas, F.M. Wahl, J. Maass, and Jürgen Hesselbach, *Towards a new concept of robot programming in high speed assembly applications*, International Conference on Intelligent Robots and Systems, IEEE, 2005, pp. 3827–3833.

[21] Markus Waibel, Michael Beetz, Javier Civera, Raffaello D'Andrea, Jos Elfring, Dorian Gálvez-López, Kai Häussermann, Rob Janssen, J.M.M. Montiel, Alexander Perzylo, Björn Schieß le, Moritz Tenorth, Oliver Zweigle, and René De Molengraft, *RoboEarth*, IEEE Robotics & Automation Magazine **18** (2011), no. 2, 69–82.

ISBN 978-3-8007-3601-0

2.10 Discussion and Conclusion

This chapter has been centred robot guidance with single camera for object pick and place tasks in industry. The chapter reviewed the common pipeline for feature based pose estimation of objects in an automation scenario. The two contribution, [Contribution B] and [Contribution C], describes the initial version of the Robot Co-worker platform that enables easy changeover between different tasks. The system presented in the papers includes a 2D, 2.5D and 3D pose estimation system named the *perception pipeline*, which detect objects with techniques described in this chapter. The contributions showed that it is possible to make a system, which is easy to re-configure and train objects in 2D, 2.5D or 3D pose estimation scenarios.

During the work with the vision integration it has become clear that making the vision system re-configurable in the same way as the robot motion control is difficult. Many of the objects that has to be handled do not have rich texture that make detection of interest points difficult. The objects are typical metal parts with few geometric features like holes, edges and other geometric features that requires other detection methods e.g. edge based or trained pattern recognition to detect distinctive geometrical image features/regions. These methods are typical associated with many specialized parameters like circle diameters and thresholds etc. which can be difficult to adjust correctly without in-depth knowledge about machine vision. These parameters makes it difficult to train the vision system to detect new objects reliable. However, this is not the biggest challenge. When the robotic work cell is changing or the calibration planes is changing e.g. due to another height of a object a new calibration is required for 2D applications. A procedure which for some people seems difficult and hard to make 100% automatic. In opposition, single camera 2.5D and 3D applications is easier to automate because only a hand eye calibration is needed. Unfortunately, these methods requires 2 or 4 distinct feature points, which for some objects is hard to find. In 2.5D and 3D pose estimation, world coordinates for the object has to be provided. Typical they have to be derived from a CAD model or a technical drawing ensure the required precision. Again, this procedure is inconvenient for the production staff because they often need to contact a mechanical engineer or people with skills in 3D CAD computer programs. One solutions to this problem is to automatic train features and extract world coordinates from the CAD model as known from geometric/edge based CAD matching.

Moreover, 2D detection techniques required that the features that must be detected it clearly visible in the image without influence from ambient illumination. In order to make a reliable robot guidance application, the correct lightning con-

ditions must be present to detect the required features. This requires knowledge in lightning setting e.g. dark/bright-field light, diffuse or back lightning. Other, solutions to the specific problem is to equip the robot with its own light source and lens filter to supress ambient light. If this is not enough multi exposure techniques like HDR algorithms could be applied to ensure high contrast images.

Creating 2D single camera pose estimation systems where shop-floor workers easily can re-program/re-configure the vision system to detect new novel parts e.g. via a GUI, requires a huge engineering effort to make system user friendly. In the end this approach of training vision systems is what the major camera producers like SICK and Cognex have tried with their smart cameras in the last 10 years. They have almost succeed as long the object is similar in size and geometry and the lightning conditions are in place. However, these easy accessible cameras are limited to 2D vision applications and are not able detect objects with a 6D transformation like objects in boxes and bins. Additional, they cannot handle large amount of occlusion because they use trained 2D patterns instead of local image features. This is a problem in many real world automation tasks.

During the last couple of years the robot manufactures have become aware of the problems that exists during integration of vision with robots. As a solution many of the Robot vendors are started to integrate vision directly with the controller in the same way as the RobotCoWorker. Their solutions are many times build on top of a OEM smart camera with additional calibration functionality to align camera and robot frame as described in Section 2.3.1. These products include, ABB Integrated Vision [14], Fanuc [15] and Motoman [16] At the robotic and automation fair, Automatica held in Germany in June 2016, the Canadian company Robotic introduce a small wrist mounted camera and 2D vision system for picking separated part from a table [17]. Their sales headline is "New robotiq vision system breaks down integration barriers". This is a clear sign that the large manufacture of robot equipment has seen the challanges and are starting to introduce similar products as the Robot CoWorker vision system.

In many automation projects dedicated vision processes and hardware are required to reliable detect each object. Many times quality control is needed as a

[14] http://new.abb.com/products/robotics/application-equipment-and-accessories/vision-systems/integrated-vision

[15] http://robot.fanucamerica.com/products/vision-software/robot-vision-software.aspx

[16] http://www.motoman.com/products/vision/

[17] http://blog.robotiq.com/new-robotiq-vision-system-breaks-down-integration-barriers

part of the procedure. In order to support these special cases the vision module in the Robot co-worker are now extended with a script environment. This new vision functionality in the robot co-worker are build around the Machine Vision library Halcon, that offers the script solution, HDevEngine[18]. HDevEngine is an engine able to execute Halcon scripts. With this engine integrated into the Robot Co-worker framework the customer benefits from the flexible and easy re-programmable Robot co-worker and have the possibility to get customized vision functionality developed by a vision expert.

A major issue in 2.5D and 3D single camera robot vision is that the third dimension are inferred from image features. In addition, objects need to have distinct visual features in order to use feature based method as described in this chapter. This is properties, which many of the industrial objects presented in the introduction not posses. If these objects has to be detected typical edge based methods have to be applied like 2D pattern matching or CAD matching. If 6D poses are needed edge based matching methods is applicable but it requires distinct edge feature, which often needs extra lightning to ensure. The conclusion of the work presented in this chapter is that 2D single camera methods are established methods and work well. However, each methods comes with a set of requirements to work properly and limitations. With 3D sensors and 3D pose estimation techniques we could avoid some of these requirement and drawbacks.

In 3D vision the third dimension is explicit given and with many of the new 3D sensors, which is pre-calibrated from the factory, the only calibration needed is a hand eye calibration before you are ready to pick objects. 3D pose estimation techniques uses CAD models as prior knowledge. Thus, there is no need for a manual training phase as the case from 2D feature-base pose estimation. CAD models provides the appropriate abstraction in order to give a easy instruction of the vision system. With 3D pose estimation the same amount of visual features is not needed compared to 2D because many 3D pose estimation algorithms relies on local shape features.

[18] HDevEngine: `http://www.halcon.com/hdevelop/hdevengine.html`

Chapter 3

3D Estimation

3.1 Introduction

One of the prerequisite for reliable 3D pose estimation is good 3D data. In 3D pose estimation applications it is important to have 3D data with minimum noise and too many missing points. In this chapter 3D estimation techniques and challenges will be discussed. A new structured light scanner suited for robot tool mounting will be presented in the contribution Section 3.6. The sensor takes up some of the problem in conventional 3D sensors such as problems with specular surfaces and inter-reflections. The review of the existing methods for 3D estimation will focus on the typical sensor modalities used in modern factory automation; including structured light scanning, active stereo and laser line triangulation. These methods are active method that all illuminate the measurement scene with artificial light. Active methods are special useful in 3D estimation of texture-less objects where limited natural cues exist on the surface. Photometric stereo techniques like Shape-from-Shading [ZTCS99], Shape-from-Silhouette [CBK04] and Shape-from-focus/defocus [NN94] are not covered in this chapter. Neither is single camera reconstruction techniques like Structure-from-motion (sfm) considered [WBG+12].

3.2 Commercial 3D Sensors - a review

The classification of 3D computer vision systems for factory automation is still not as mature as known from 2D. Simply because the sensors and smart devices are still missing. In the later years we have seen some products entering the marked as 3D smart cameras but the amount is still limited and the prices are high compared to 2D vision. The main benefit of 3D smart cameras is that they are pre-calibrated for a given working range, which enables seamless integration. One of the few vendors that offers real 3D smart cameras where the data processing runs onboard is SICK that released the SICK Trispector[1] in fall 2015. This camera is a laser line sensor, which provides simple on-board 3D methodology tools and a tool for matching simple shapes like rectangles and ellipses. The sensor requires motion in order to reconstruct the surface by using sheet-of-light techniques. The predecessor of SICK Trispector, is the SICK IVC-3D but was not a sales success due to the cumbersome programming and price. The Canadian company LMI Technologies released in 2015 the first structured light snapshot smart camera. The LMI Gocator 3100 series[2] uses two cameras and a blue LED projector to project fringe patterns. The sensor is made for close up 3D with a maximum distance to the object of 100 mm.

Traditional 3D technology in the automation industry mainly have been dominating in the quality control domain where laser triangulation sensors often are used. Laser triangulation sensors are sensors that measures the deflection of a laser line and compute 3D points using triangulation techniques. Sometimes this type of sensor is referred to as profile sensors. The sensor requires that either the object of interest or the sensor is moving in order to create a depth image. Many different sensors are available on the marked today and this sensor technology is considered as a standard product. Laser triangulation sensors exist in both a pre-calibrated single unit where camera and laser are integrated in a single unit and stand-alone-camera versions where the laser have to be added before having a running application. Commercial stand-alone-cameras include e.g. SICK Ranger [3] and Automation technology C2/C3/C4 [4] among other. These sensors provide the maximum flexibility to change the field of view and baseline. If this flexibility is not needed and the object size, distance to the sensor and required accuracy are known in advance it can be an advantage to choose a pre-calibrated single unit sensor. Examples of these sensors are SICK

[1] https://www.sick.com/us/en/product-portfolio/vision/3d-vision/trispector1000/c/g389052

[2] http://lmi3d.com/products/gocator/snapshot-sensor#

[3] https://www.sick.com/us/en/product-portfolio/vision/3d-vision/c/g138560

[4] http://www.automationtechnology.de/cms/en/

Ruler [5], LMI 2100/2300 [6], Automation Technology C5-CS [7], Leuze LPS 36 [8] and Cognex DS1000 [9]. Laser triangulation sensors with two cameras are available in applications where self-occlusion from the objects have to be a minimized. In cases where objects have protruding shapes a sensor like LMI Gocator 2880 [10] with two cameras is a better choice, than a conventional sensor with one camera. The SICK Scanning ruler belong to a new class of laser triangulation sensors, which has a motorize laser is embedded in the sensor such that the sensor sweeps the laser across the scene. The sensor is intended for scanning of euro pallets in bin-picking applications.

During the last 10 years, pre-calibrated stereo cameras have entered the marked as single plug n play units. These cameras are found as industrial grade cameras e.g. Point Gray BumbleeBee cameras [11] and Scorpion Stinger 3D [12] and consumer grade cameras as the Zed stereo camera [13]. Recently, IDS Imaging introduced the Ensenso [14] stereo camera with a small integrated projector to illuminate a scene with a salt/pepper noise pattern to accommodate the need for 3D imaging of non-textured objects.

In applications where high accuracy is required a few commercial structured light sensors suitable for industrial environments are available. Most of the sensors are fringe projecting sensor with one camera. The sensors have a restricted working range, which is very limited. Sensors like the VrMagic area scan [15] are restricted to a working range of 50 mm and the ShapeDrive [16] sensor is limited to 300 mm in depth. The sensors are highly accurate in their working range but better for quality inspection tasks of small objects than robot guidance. One of the problems with these sensors is that they do not handle projector defocus. A problem that is handled in the proposed 3D sensor in Contribution C in Section 3.6. In 2015, the ShapeCrafter3D [17] sensor was released, which are taking up the challenge. This sensor has a larger working range and a very good accuracy

[5] https://www.sick.com/us/en/product-portfolio/vision/3d-vision/c/g138560
[6] http://lmi3d.com/
[7] http://www.automationtechnology.de/cms/en/
[8] http://www.leuze.com/en/deutschland/produkte/messende_sensoren/3d_sensoren_1/lichtschnittsensoren_1/index.php
[9] http://www.cognex.com/products/machine-vision/ds-1000-displacement-sensor-laser-profiler/?id=13693&langtype=2057
[10] http://lmi3d.com/products/gocator/profile-sensor
[11] https://www.ptgrey.com//bumblebee2-firewire-stereo-vision-camera-systems
[12] http://www.scorpionvision.com/
[13] https://www.stereolabs.com/
[14] https://en.ids-imaging.com/ensenso-stereo-3d-camera.html
[15] https://www.vrmagic.com/imaging/3d-sensors/
[16] http://www.shape-drive.com/index.php/ShapeDriveHome.html
[17] http://www.shapecrafter.no/index.php

Figure 3.1: Upper left: Commercial laser triangulation sensors **Upper right:** Commercial structured light sensors **Lower left:** Commercial pre-calibrated stereo sensors **Lower right:** commercial time-of-flight sensors

for robot picking applications at around 0.05 mm in a typical working distance. The marked for 3D sensors suited for robot picking applications is still very limited. In general most of the structured light scanner on the marked are in the category of high-end sensors aimed at meteorology or reverse engineering applications in non industrial environments.

One of the advantage with the high-end structured light sensors is that they are typically optimized to reconstruct reflective surfaces. Many application for these systems involve making 3D scans of fabricated metal parts like car engines or milled metal parts. However, the sensors are physical large, expensive and not suited for an industrial production environment. These structured light sensor are typical digital fringe sensors that apply two or more camera. The application domain for these sensors is typically reverse engineering, sample-

based geometrical quality control as metrology task where an object surface is compared to a CAD model. This high-end metrology market is dominated from a few large companies like Gom[18], Zeiss Optptechnik(previous Steinbichler Optotechnik GmbH)[19], Kreon Technologies[20], Creaform[21], Aicon3d (previous Breuckmann)[22], Nikon Metrology[23] and Geomagic[24]. All these companies develop and market their own 3D scanners. Their 3D scanners are either hand-held laser line or stationary structured light scanners. Most of the 3D digitizers are delivered with dedicated 3D metrology software to analyse the 3D scans and/or create high resolution 3D models of the scans.

Until 2010 where the Kinect 1 was released Time-of-flight (ToF) sensors were considered as an alternative to stereo vision in both commercial applications and research. After 2010, the use of time of flight cameras in robotic applications has dropped significantly. However, the marked for commercial ToF cameras are still existing and even new cameras have entered the marked in the last year. Large companies line Basler and SICK have introduced their own industrial ToF camera accommodating the new for "Kinect like" sensors in industrial grade. The 3D ToF from Basler[25] is a VGA camera with an accuracy of +/- 10 mm and the SICK 3vistor-T[26] has a image size of 174x144 pixel with an accuracy of+/- 3mm at 1 meters distance. With the introduction of the SICK 3vistor-T we start to see ToF cameras with the required accuracy in order to use it in 3D robot picking applications. With the newly introduced Kinect 2 ToF camera a promising future for low cost ToF cameras has started where 3D sensors are cost efficient and build into autonomous robots. Traditional different brand like Swiss Ranger[27], Odos[28] and PMD CamCube offers of the shelf Time-of-Flight cameras.

[18] http://www.gom.com/
[19] http://optotechnik.zeiss.com/en/
[20] http://www.kreon3d.com/
[21] http://www.creaform3d.com
[22] http://aicon3d.com/start.html
[23] http://www.nikonmetrology.com
[24] http://www.geomagic.com/
[25] http://www.baslerweb.com/en/products/cameras/3d-cameras/time-of-flight-camera
[26] https://www.sick.com/us/en/product-portfolio/vision/3d-vision/3vistor-t/c/g358152
[27] http://hptg.com/industrial/
[28] http://www.odos-imaging.com/

3.3 The projective camera model

In this section camera models, camera calibration and epipolar geometry are outlined, before continuing with the different 3D estimation techniques. For defining the camera geometry we need to make a model of the camera, which is describing the transformation from a 3D point in the scene to the 2D point at the image plane. The common camera model today is the pin hole or projective camera model. The ideal projective camera model assumes that all light rays passes through only one point, the focal point before hitting the image plane. The projective camera model projects the 3D homogeneous world coordinates $\{x, y, z, h\}^T$ of a point in the scene to the 2D homogeneous image coordinates $\{x, y, h\}^T$. The model includes the intrinsic parameters $\mathbf{A}$ in Equation 3.3 and the extrinsic parameters $\mathbf{[R \mid t]}$ in Equation 3.3. The intrinsic matrix $\mathbf{A}$ contains the *focal lenght f*, which is the distance from the optical point of the camera lens to the image plane. Changing the focal length is therefore equivalent to zooming. The *optical center* $\{\Delta x, \Delta y\}$ models the point where the optical axes intersect with the center point of the image plane with respect to the top left corner, normally. In an ideal world the optical center will be in the middle of the image plane but due to imperfect production of the image sensor and distortion of the lens this point can vary a little. α and β models the affine transformation, which each pixel has due to production inaccuracy of the image sensor. If each pixel square at the image sensor is quadratic $\alpha = 1$ and $\beta = 0$. The extrinsic parameters $\mathbf{R}|\mathbf{t}$ in Equation 3.3 are the rotation and translation needed to project the 3D world coordinate onto the 2D image frame under an ideal projective camera model where the intrinsic matrix $\mathbf{A}$ is equal an identity matrix $\mathbf{I}$.

$$\mathbf{P}=\mathbf{A}\left[R|t\right]; \quad \mathbf{A}=\begin{bmatrix} f & f\beta & \Delta x \\ 0 & \alpha f & \Delta y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.1}$$

With the projection matrix $\mathbf{P}$ defined, a homogeneous 3D world coordinate $Q_{world} = [x, y, z, h]$ is easily projected to the 2D image plane by calculating $q_{image} = \mathbf{P}Q_{world}$.

The projective camera model does not account for radial and tangential distortion introduced by the optic on a camera. Even through the manufacturers of lens optics and image sensors aim to avoid adding distortion to the image, in practice the lens adds radial distortion and the skewed mounting of the image sensor introduces tangential distortion. The projective camera model presented in Equation 3.3 does not account for the non-linear radial distortion introduced by the lens, Figure 3.2. For more details see [HZ04].

In an orthographic camera model, the world points are simply translate parallel

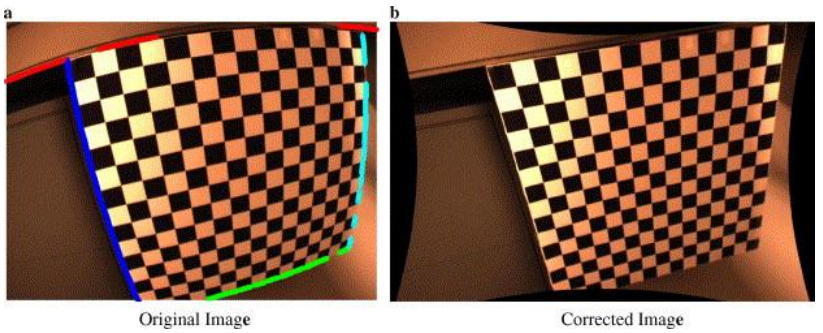


Figure 3.2: Example of radial distortion

to the optical axis to map from world to image coordinates. The projection matrix for this camera model is equal to Equation except that **A** equals the identity matrix. This model is valid for cameras with telecentric lenses with no perspective distortion.

When the camera model is defined the camera calibration process is the procedure to estimate the projection matrix. Lens parameters k are often computed in a separate non-linear optimization process. In traditional camera calibration a set of calibration images with a calibration target in the image are taken. The calibration target is a planar object with easily detectable and known features. A calibration target has typically a chess or grid pattern, which is easy to detect with known image processing techniques. The calibration target defines a world coordinate system such that the 3D feature coordinates are known in advance. With a set of known calibration images taken of a known calibration target, the task of camera calibration is to solve Equation 3.3 with A as unknown. The extrinsic parameters $(R|t)$ are solved from the known 2D/3D point correspondences and e.g. techniques described in Section 2.6.2. How to solve the problem depends which method is used.

One of the early methods for camera calibration is [Tsa86]. Later, the paper from Heikkilä and Silvén[HS97] included more intrinsic parameters. A modern

approach for camera calibration was proposed by Zhang [ZTCS99],[Zha00]. This method is a part of the popular computer vision library OpenCV [29] and is the method used during this Ph.D. Camera calibration is not a part of this Ph.D, hence the topic is not covered in-depth. For a comprehensive review of existing camera calibration techniques and the accuracy, the reader is pointed to [QLZ10] and [SAB02].

3.3.1 Epipolar geomerty

Estimation of depth with triangulation techniques requires informations for minimum two view points, which observe the same point in the scene. In binocular vision, two cameras are used to relate one point seen in the first camera to one in the second camera. Likewise, laser triangulation sensors and structured light sensors which consist of one camera and one illumination source, apply the same technique to compute depth from the deflection of points or lines projected into the scene. In this section, the epipolar geometry is covered.

Epipolar constraint is fundamental in stereo matching algorithms to reduce the search problem in stereo matching from a two dimensional to a one dimensional problem. Adding this constrain to a stereo matching algorithm requires that the epipolar geometry of the stereo setup is known. To derive the geometry between two cameras viewing the same 3D world point $\mathbf{X}$ as in Figure 3.3, we need to establish a common world coordinate frame.

The location of this frame is not important. It could be located at both the left or right image plane or even between the two cameras. In robotic vision it is sometimes even beneficial to locate the common world frame at the robot base to make the robot control easier. Then the 4x4 transformation matrix $\mathbf{[R \mid t]}$ is needed to relate one of the image planes to the robot base. To lower the mathematical complexity we chose to set the left camera coordinate frame as the world frame, which results in the two camera matrices.

$$P_l = A_l \left[I|0\right] , \; P_r = A_r \left[R|t\right] \tag{3.2}$$

The overall goal of deriving the *epipolar geometry* is to construct a model, which relates the projection of a 3D world point into the first image plane to the projection of the same point into the second image plane. From Figure 3.3 the

[29] `http://opencv.org/`

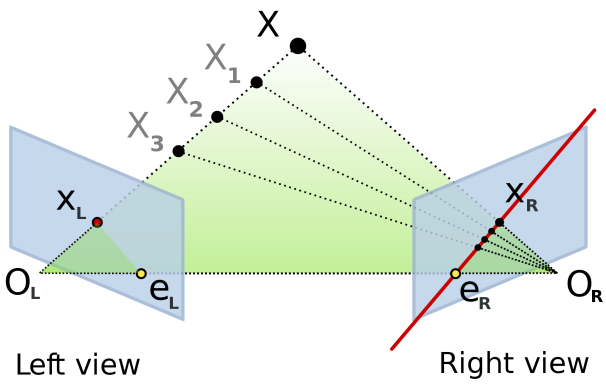


Figure 3.3: Illustration of the epipolar geometry

projection of the 3D point X projected to the left image plane constructs a virtual line between X and the optical center of the left camera O_L. Let us denote this virtual line L_v from the left camera point of view the 3D point X will be on this line but the distance is unknown. The 3D points X, X_1, X_2, X_3 will be projected to the same 2D point at the left image plane. The projection of the line L_v into the right camera image plane will be a line. This line is the *epipolar line*. When the epipolar geometry is mathematical known the epipolar line in both images are calculated utilizing triangulation to aid the stereo matching algorithms in the search for matches between the two views. To start formalize the mathematical derivation of the epipolar geometry we note that the two optical centres of each camera $\{Q_l, Q_r\}$ and the 3D world point X point lie on a plane; the epipolar plane.

The epipoles in epipolar geometry refer to the point projection of the other camera optical center Q_r on to the image plane point. The epipole is located outside the image planes when the two cameras are parallel. If the cameras are tilted towards each other the epipols will move from the edge of the image plane toward the optical center. The pose of the epipoles will only be located at the optical center in the theoretic camera setup where both cameras point towards each other. The essential matrix is given in Equation 3.3.

$$E = A^T F A \Leftrightarrow E = R[T]_x \tag{3.3}$$

where **E** is the Essential matrix, **A** is the intrinsic matrix and R is the rotation matrix, **T** is the translation vector between the camera positions and $[\mathbf{T}]_\times$ is

the 'cross product matrix' of $\mathbf{T}$.

The essential matrix $\mathbf{E}$ contains information about the translation and rotation between the cameras in the camera coordinate frame. The fundamental matrix $\mathbf{F}$ contains the same information as the essential matrix plus the intrinsic parameters for each camera. The fundamental matrix relates the two cameras in pixel coordinates. Essential, the essential/fundamental matrix can provide the epipolar line to search along in the right view by multiplying a point coordinate in the left image $\mathbf{x_l}$ with essential/fundamental matrix. The Fundamental matrix is given in Equation 3.4.

$$x_r^T F x_l = 0 \quad \Leftrightarrow F = A^{-T} E A \quad \Leftrightarrow F = A^{-T}[T]_x R A \tag{3.4}$$

where $\mathbf{x_l}$ and $\mathbf{x_r}$ is the pair of corresponding points in left and right image and $\mathbf{F}$ is the fundamental matrix, $\mathbf{A}$ is the intrinsic matrix and $\mathbf{R}$ is the is the rotation matrix, $\mathbf{T}$ is the translation vector between the camera positions and $[\mathbf{T}]_\times$ is the 'cross product matrix' of $\mathbf{T}$.

In stereo calibration, the extrinsic parameters $[\mathbf{R}|\mathbf{t}]$ of a stereo pair are estimated. Before a stereo calibration each cameras intrinsic parameters must be known. With both the extrinsic and intrinsic parameters it is easy to compute the essential and fundamental matrix from Equation 3.3 and 3.4.

3.4 3D Estimation of industrial objects

Despite the fact that many different techniques for reconstructing a scene exist, not all are suited for industrial production where objects are often without texture and/or have non-lambertian surface properties. Furthermore, if a robot with a camera mounted in either the tool or as static scene camera is considered, typically it is a too hard a constrain to require motion. This is because of three reasons; First, objects on a table or in bins/boxes are not moving and secondly, techniques that require motion e.g. Structure from Motion needs good features to track, to estimate the camera position and reconstruct the scene structure. Third, the required time for moving a sensor is not always applicable in order to decrease the cycle time of the robot. In a production environment/scenario where flexible robots and picking solutions are required, it is typically not feasible to enclose the scene e.g. light sources, which is required for shape from shading and photometric stereo techniques. Furthermore, shape from shading and photometric stereo techniques are today mainly applied in fine grain surface inspection of objects to find small scratches and other defects in the surface. All

the mentioned constrains for a general 3D sensor for robotic, imply that only techniques as structured light, stereopsis, time-of-flight and laser triangulation are practical suited for acquiring point clouds in robotic picking scenarios. Based on these techniques we will list some of the problems that exist today, which makes the techniques non-optimal.

In this section an overview of the different method will be presented and the pros and cons for the different methods are outlined. An overview of the early development until 2004 in this area is given in [Bla04].

3.4.1 Laser triangulation

Laser triangulation techniques reconstruct a scene by sweeping a laser line and compute the depth based on the line deflection. The technique requires movement of either the object or the laser line to be able to accumulate line profiles to a full 3D surface. Laser line triangulation suffers one major drawback as 3D sensor methodology for robotic picking applications, as the method requires movement of the object or the laser line. For static scenes it can be resolved by attaching the laser beamer to a motor that performs the movement as e.g. the SICK Scanning Ruler sensor. Furthermore, the laser line is highly influenced by the surface properties, where highly reflective surfaces are scattering the laser beam and dark objects are absorbing the laser beam. The scattering effect can be reduced by applying different laser colors with wavelength that is less absorbed by the surface. Changing laser wavelength forces you to shift the camera band pass filter to the right wavelength, which makes the technology less appealing as a general sensor technology. The nature of lasers results in limited range where the laser is in focus which reduces the depth of field for the sensor. This can however be corrected to some extend in the line detection algorithm by estimating the middle of the laser line by doing statistical analysis of the gauss distribution created by the laser beam on the object. A wider line, results in a wider gauss distribution. The geometry of the laser camera setup is crucial parameters that determine the resolution and working range of the setup. A rule of thumb is that a lower triangulation angle and baseline results in lower resolution. In classic laser triangulation one camera and one laser stripe projector are used. This setup can cause the object to self-occlude due to the triangulation angle between camera and laser which is not desirable. A common solution to the problem is to utilize two or more cameras. In general laser line triangulation is an accurate and robust method for estimating 3D surfaces but the sensor setup has to be configured for the application. However, the method suffers from the limited depth of field, the laser scattering effect on shiny objects and the fact that the object or laser line have to sweep the surface.

3.4.2 RGB-D

Since the introduction of the Microsoft Kinect sensor in November 2010 and later Kinect One in April 2016, new RGB-D sensors have drawn high attention in both the computer vision and robotic research community. While the Kinect sensors originally were designed as a gaming interfaces, they have interesting perspectives because of the low price and the a decent quality depth sensor. In this section, a exposition of this special class of low cost 3D sensor is covered. Throughout, this exposition we will name the first kinect as Kinect 1 and the recent released version as Kinect 2.

The Kinect 1 sensor is an active sensor, which consists of a VGA RGB camera, an ir projector, an IR sensor, tilt motor and a microphone array. The depth sensing system consists of the structured light IR projector, which projects a known IR pattern of dots towards the scene. With the integrated IR sensor the Kinect measures the shift of the IR pattern when it hits an object. This is achieved by comparing the deflected pattern to a known pattern stored in memory utilizing a small 9x9 or 9x7 correlation window[30], [KE12]. Both the depth sensor system and the RGB camera run 30 frames per second, which is sufficient for most robotic and computer vision application, [CLV12].

The Kinect 1 sensor has already been applied in various application within robotics and computer vision. [IKHM11] and [NIH11] developed the system KinectFusion to make a full dense 3D reconstruct of a scene in real time. The work by [NIH11] includes a dense surface mapping of the scene by fusing the live depth map from the kinect to the already obtained surface. This surface mapping is done using a signed distance function, [CL96]. Registration of new depth maps to the surface requires camera tracking. A course-to-fine ICP algorithm is calculating the 6 DOF camera movement between frames, [BM92],[RL01], [PMC+11]. [IKHM11] implemented the KinectFusion on a GPU using generic programming. In general there has been a lot of research with the kinect sensor concerning 3D mapping and modelling application during the last 5 years.

Work by [KE12] presented a study of the accuracy and depth resolution of the Kinect 1 sensor. [KE12] compared the point cloud from a calibrated Kinect with a high-resolution Faro LS 880 3D laser scanner to investigate the system-

[30] `ROSKinectcalibration-http://www.ros.org/wiki/kinect_calibration/technical`

atic error and the resolution of the Kinect depth measurement. The reference laser scanner has a nominal range accuracy at 0.7mm at 10m distance [31]. The average point resolution of the Faro LS 880 point cloud on a surface perpendicular to the range direction is 5 mm. The systematic error for the Kinect 1 was computed by register the Faro LS 880 point cloud and the Kinect 1 point cloud using three different ICP methods. When it is ensured that the registration is correct, 1000 points in the kinect point cloud is randomly selected. For each of these points the nearest neighbour is calculated in the Faro LS 880 point cloud. The mean distance between the points was computed to 0.1, 0.0, 0.1 cm in the x,y,z-direction, respectively. This indicates that there is no significant systematic error in the Kinect point cloud. The same measurement was conducted without any calibration of the Kinect, which resulted in a mean distance of -0.5, -0.6, -0.1 cm in the x,y,z-direction, respectively. This indicates that additional calibration of the kinect besides the internal is needed if a low systematic error is required. The resolution of the depth data is measured using plane fitting test, where the Kinect measures the distance to a planar door. Ten measurements are conducted in the Kinects operation range from 0.5 - 5.0 m. For each measurement the same area of the door is selected in the point cloud and fitted to a plane using a RANSAC plane fitting method to avoid influences of outliers. To evaluate the random error of the depth, 4.500 points at the plane were randomly selected and the standard deviation was calculated.
The conclusion of the test showed a decreasing depth resolution where the depth accuracy of the Kinect 1 is low and approximately *7 cm* at the maximum range of 5 meters. Whereas, the accuracy at 3 meter is approximately *2.5 cm* and at 1 meter quite small; approximately *2 mm*. For applications where the measurement distance is within 1-3 meter i.e. mobile robot mapping the resolution is acceptable but the quality of the depth measurements degrade by noise and low resolution when the distance increases. In applications where high accuracy is needed like robot picking, 2 mm is acceptable but in general it is desirable to have a better accuracy. Mainly, because the error propagation in a robot system can exceed the tolerances in the system, which in the end will cause a failed pick.

In many 3D robot picking applications it is often an advantage to use several 3D sensors to cover the scene and reduce occlusion. With the in-expensive kinect sensors it is extra attractive but unfortunately the Kinect sensor suffers from interference when using more than one camera, because the Kinect is projecting identical IR patterns. However, Butler *et al.* [BIH+12] proposed the "Shake'n'Sense" approach where each Kinect in a multi-sensor setup is continuously shaken using an imbalanced rotating motor. Then the projected pattern will appear significantly blurred for another device due to the high frequency motion that the motors apply. Another, solutions to the problem is to physical

[31] Faro homepage - `http://www.faro.com/focus/uk`

time-multiplex the sensors with a hardware shutter that physical plock the pattern projector [BRS+11].

In April 2016, the newest kinect 2 was released. The sensor is a Time-of-flight device, which utilizes a Continuous Wave (CW) Intensity Modulation approach. This approach is the most commonly used in ToF cameras [SLK15]. The sensor actively illuminate a scene using near infrared (NIR) intensity-modulated periodic light. During the travel of the illumination beams a time shift is caused in the optical signal, which is equivalent to a phase shift in the periodic signal. The phase shift is detected in each pixel of the ToF image chip in a so-called mixing process. The sensor-object distance is easily estimated from the phase shift as the speed of light is known and that the light has to travel the distance twice.

In [SLK15] a comparison of Kinect 1 and Kinect 2 was presented. The comparison showed that the multi-device interference is lower for Kinect 2 compared to Kinect 1. However, there exists periodic interference error such that the Kinect 2 some times has a high depth measurement error. The Kinect 2 sensor is better to suppress ambient light compared to Kinect 1 but with the cost off a high measurement error. Hence, Kinect 1 outputs precise point measurements under moderate ambient light condition but fails in reconstructing points when the ambient illumination increases. Furthermore, the studie from Sarbolandi *et al.* [SLK15] relieved an interesting fact. The evaluation showed that Kinect 1 is better to handle reflective objects than Kinect 2 due to the different technology but Kinect 2 is better to model a plane without deflection of the corners of the plane.

In all, the two Kinect sensors are in-expensive 3D sensors, which have many application areas. However, for precise 3D Robot picking none of them is an attractive sensor due to the need of accuracy, the general noise level and inability to reconstruct reflective industrial objects. In robot applications where reflective objects are not an issue and the general required accuracy is low, the RGB-D technology could be an in-expensive choice. These applications could involve picking of textured objects where precision grasp is not a requirement. An example of these applications and a picking solution are presented by the Belgium company Intermodalics [32].

[32] `http://www.pickit3d.com/en`

3.4.3 Stereo vision

Computational stereo refers to the science of perceiving depth when a scene is perceived with a binocular vision setup. The techniques of measuring depth from two or more images acquired with different viewpoint have been a heavily studied research area since [Rob63] introduced the technique of stereo triangulation in his Ph.d. thesis from 1963. Calculating the depth from two or more images requires that points in one images can be related or matched to a point in the other images with a desired accuracy and without ambiguous matches. Thus, it is required that good distinct features in each image exist to be able to find correspondences between two images. If good features in the scene exist, corresponding matches are found by searching for the best candidate point along the epipolar line. When corresponding pairs are found we can compute the disparity. In dense stereo we compute a disparity map where each pixel correspond to the disparity. With the disparity map, the baseline and the focal length given, triangulation computes the position of the correspondence in the 3D space. For stereo configurations where the cameras have parallel optical axis the depth is estimated from the principle in Equation 3.5

$$z = \frac{b \cdot f}{d} \tag{3.5}$$

where **b** is the baseline, **f** is the focal length and **d** is the disparity. This results in a discretization of the depth that is determined by the camera resolution and the epipolar geometry. A way to increase the depth accuracy is to apply method for computing the depth based on subpixels.

With stereo vision an object like a cereal box will typical result in a good reconstruction due to the many features on the surface while a reconstruction of a metallic cylinder will fail. In the search for better performance lots of the proposed work in stereo matching have projected artificial texture onto the scene like random dot patterns. This increases the reconstruction performance considerably in terms of number of reconstructed points. Since [Nis84] proposed the used of a texture projector to aid stereo matching algorithms, some applications have been created, which benefit from the extra texture added to the scene. Kang *et al.* [KWZK95] used multibase line stereo to achive an accuracy under 1 mm in the area from 1.5 to 3.5 m from the cameras. The active illumination results in a denser depth map because of improved local discrimination and hence correspondence. This reduces the risk of false positive matched in the stereo algorithm. However, [KWZK95] observed that the largest errors occurred in regions where the active projected pattern not provided sufficient texture to ensure a good matching. This implies that good temporal patterns is necessary unless the object is known and the pattern can be constructed to

the particular object. The analysis by [KWZK95] shows that a frequency modulated sine wave pattern is a good solution because it does not require a large dynamic range. Randomly frequency modulated sine wave gives the best matching result because there only exist a vanishingly small probability that the same pattern will occur twice in the search area. [Kos96] showed how color patterns could increase rainbow like color pattern and using chromatic Block Matching algorithm to used the color encoding best in the matching. They showed that introducing coloured light patterns increased the amount of 100% correct matched pixel from 7.9% to 62%. This result is of cause highly influenced by the proposed block matching algorithm. Nevertheless, this early result implies that active illumination can increase the performance of stereo matching. Davis *et al.* [DNRR05] took stereo with pattern to a new level with their Spacetime stereo method. This method reconstruct a scene by considering the temporal changes from unstructured illumination changes in the scene. The method was extended to dynamic scenes by Zhang *et al.* [ZCS].

In robotic, projecting random patterns are often used. Konolige *et al.* [Kon10] investigated which random dot pattern best suited for a standard block matching algorithm. In [SKSB10] the authors used the setup from [Kon10] to learn articulation models of doors such that a PR2 service robot could learn to open doors. In [Lim09] symmetric non-recurring De-Bruijn sequences are used to improve the stereo matching performance for block matching and spacetime stereo algorithms.
Despite, good results from adding additional patters, stereo reconstruction techniques still suffer with noise and measurement errors. Noise and errors that are mainly due to insufficient accuracy in the matching algorithm even if the reconstruction is within sub pixel accuracy. The noise has different effects on the resulting point cloud and is typical bubbling or waved surfaces, wrong reconstructed points at surface discontinuities and wrong reconstruction at concave steep edge.

The literature in stereo matching algorithm is comprehensive and not in the scope of this Ph.D. The reader is guided to some of the later reviews and evaluations on the topic by Scharstein *et al.* [Sch], Brown *et al.* [BBH03], Seitz *et al.* [SCD+06], Jensen *et al.* [JDV+14], Scharstein *et al.* [SHK+] and Aanæs *et al.* .

3.5 Structured light scanning

Structured light scanning covers 3D estimation methods, which add artificial light patterns to the scene to encode the scene. Different patterns as graycode

bands, sinusoidal fringes, grids, etc. are projected on the surface to actively use the encoded images to find correspondences. The scene is taken by one [SS03], two [Gue00], three [HPS+14] or many cameras [WSRK11] placed in a known position to calculate the triangulation [SFPL10]. Even methods using two projectors and one camera have been proposed [FSDK11]. A general review and introduction to the topic is given in [SFPL10], [Zha10], [Gen11] and early work is presented in[BMS98].

Structured light estimation techniques are an appealing technology as snap shot 3D sensor technology. In comparison with laser triangulation sensors where objects have to be in motion, structured light improves the acquisition speed. The technology is now mature in terms of speed of the pattern projection and reconstruction and robustness, which makes it appealing for precise robotic picking applications where low cycle times are required [BKZ14]. However, if structured light sensors have to be applied in robot guidance applications where the sensor is mounted in the tool of the robot or above the boxes and bins, the sensors need to have a physical smaller size than general available. The need for smaller sensors becomes clear in e.g. real bin picking scenarios where objects have to be picked in the corners of a bin. Without a small and narrow 3D sensor design, the robot is not able to pick parts in corners without colliding with the bin walls.

The accuracy of a structured light system is as for all triangulation sensors dependant on the physical setup parameters like baseline, resolution of the image, sensor chip size, pixel size, lens and projector resolution. In general the accuracy of a typical structured light setup is in the microns [GIV10]. Precise calibration of the sensor is an prerequisite of an accurate sensor. Eiríksson *et al.* [EWPA16] presented a study on how the mentioned design parameters influence the accuracy and affect the overall performance of a structured light system.

With the introduction of the Kinect 1 and Primesense[33] sensors, structured light sensors have become applicable in robotic applications. Unfortunately, the sensor is not robust and precise enough for industrial robot applications. Especially, its performance with reflective object is very poor and the casing of the sensor is not robust. Nevertheless, these sensors are the first step towards general purpose 3D picking technology for robots where only a 3D sensor is mounted on the robot and the robot is autonomously taught which objects to pick.

Despite, previous advances in structured light scanning, the technology is still lacking in especially two areas. One, the robustness and quality of the re-

[33] `https://en.wikipedia.org/wiki/PrimeSense`

construction of reflective objects is not general sufficient. Techniques as high dynamic range scanning have been introduced, but challenges in computing the optimal exposure timing fully automatically and fusing the result properly still exist. Furthermore, better techniques to increase the robustness towards pattern defocus as well as separating local and global light components, which in the end have to reduce sub-surface scattering between to reflective surface patches are needed.

Since early 1980's, investigation of 3D reconstruction methods with structured light scanning have facilitated advances in many different methods. A large amount of coding strategies have been proposed, which uses both spatial and temporal patterns for dense reconstruction of static and dynamic scenes [SKKF11], [FSDK11]. In this section a review of common temporal binary and phase shifting methods is given. Methods applying spatial one shot color and grid patterns like e.g. [SBM98],[SKKF11] are not covered. Recent methods for single fringe pattern 3D shape measurments are given in [SFK14].

3.5.1 Binary encoding

The simplest coding method for structured light scanning is Binary codes. Binary codes are original proposed by Posdamer and Altschuler [PA82]. With Binary codes the scene is illuminated by a set of n temporally encoded patterns of black/white bands. Each temporal pattern is progressively halved in width and n images are captured. With Binary codes each point in the scene posses a unique binary code that differs from any other code in the scene. Depending on the number of cameras and accuracy required for the sensor, the patterns can be displayed both horizontal and vertical. In principal it is enough to encode the scene with vertical patterns. However, to increase the accuracy in especially systems with one camera and one projector, two projection directions can be applied [WOL14]. With the horizontal dimension encoded and a known epipolar geometry from a calibration process triangulation of each 3D point is computed to reconstruct depth. When the epipolar geometry is known it is a matter of searching for correspondences along the epipolar line to identify matching codewords. Some of the shortcomings with Binary codes are that even in presence of only small noises, the encoding strategy could potential generate severe encoding errors. The reason is that the brightness boundaries of the multiple patterns are in the same positions, see Figure 3.4 (uppermost). In order to overcome this Inokuchi *et al.* [ISM84] introduced Binary Gray code. The advantage of Binary Gray Code is that two neighbouring codewords have a Hamming distance of one. The fact that neighbouring columns in the projected sequence only differ by one bit makes binary gray code more robust than Binary code towards noise, which results in less reconstruction errors and spurious points. Binary coding

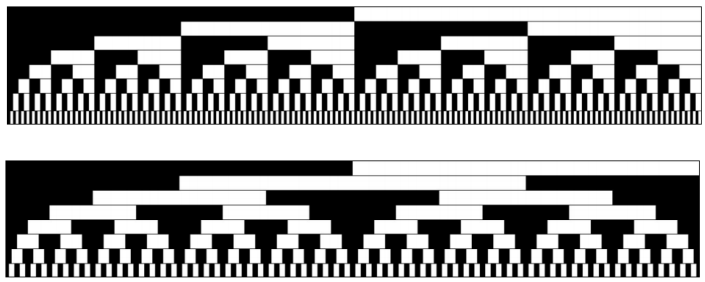


Figure 3.4: Topmost: Binary code. **Lowermost:** Binary Gray code.

strategies are in general reliable reconstruction technique and not sensitive to surface characteristics as color because they only use binary intensity values. Often, inverse pattern are used to increase the robustness towards ambient light components. In this case the difference between the normal projection pattern and the inverse are considered to determine edges during the encoding phase. In some domains and application it is problematic that a large number of sequential patterns are needed to achieve a high spatial resolution and the fact that all objects in the scene have to remain static [Gen11].

When projecting patterns at low illumination intensities, the signal-to-noise ratio of the system decreases. This induce that depth from low reflective regions cannot be obtained. Opposite, if high illumination intensity patterns are projected, standard structured light algorithms have problems in estimating the depth from regions with high reflectance due to pixel saturation. Thus, most binary coded structured light techniques assume that the objects have uniform albedo, otherwise, the whole surface cannot be reconstructed.

In 2000, Gühring *et al.* [Gue00] introduced the Line Shifting method, which is a combination of Binary Gray code and phase shifting methods. The integration of the two methods bring together the advantages of both coding strategies. Line shifting benefits from the robust decoding in Binary Gray code without ambiguities as known from Phase Shifting and the high resolution from Phase Shifting Methods. This results in highly accurate 3D reconstruction but with a higher number of projected pattern as a consequence [SFPL10]. The idea of Line shifting is to combine temporal gray code and the idea from laser line triangulation where a line is sweeping across the scene. In Line shifting, one line is in principle enough to reconstruct a scene. However, with a digital projector multiple lines can be projected in parallel to increase the speed. In the same way as for laser line triangulation, correspondences are found by detecting the

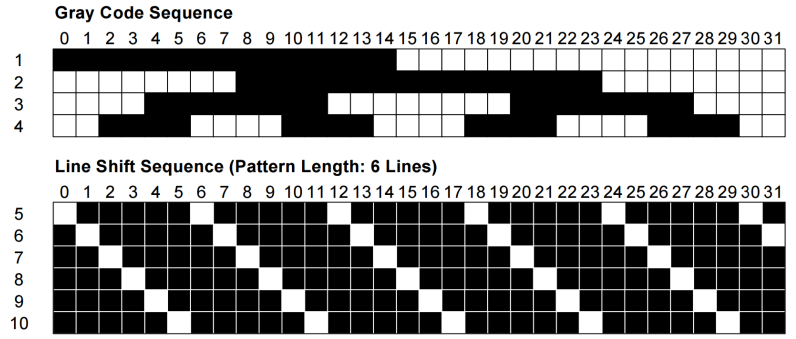


Figure 3.5: Topmost: Binary Gray code. **Lowermost:** Phase shifting code (Line shift). The sequence is shown with $n = 4$

peak of the projected stripe, line shifting locates the peak of each line. Several methods for detecting the peak with sub-pixel accuray exist including different filtering methods [NF91], spacetime analysis [CL95]. Projecting several lines in parallel create ambiguities. In order to resolve this ambiguity a binary gray code pattern is projected before the line shifting patterns are projected, to label each line and determine uniquely the line number. This coding strategy is applied in [Contribution F] where it is used to scan scenes of objects to create a pose estimation dataset.

The presented methods for binary encoding a scene are the most common coding strategies, which all requires a larger amount of temporal projected pattern. Work has been proposed where several intensity levels are used to lower the amount of projected patterns. These methods are known as n-ary coding strategies and to well-known methods includes [HK97] and [CKS98]. The details of these methods are not covered in this thesis.

3.6 Contribution

Paper 3: *A Structured Light Scanner for Hyper Flexible Automation*

[Contribution D], entitles "*A Structured Light Scanner for Hyper Flexible Automation*" is published on the 2nd International Conference on 3D Vision held

from the 8th to 11th of December 2014 at the University of Tokyo, Japan.

The paper presents a small structured light sensor, which implement novel techniques to lower inter-reflection, sub-surface scattering and projector defocus. A video of the system running is found at the Danish Technological Institutes youtube channel: `https://www.youtube.com/watch?v=0aBM31XZjDw`.

2014 Second International Conference on 3D Vision

A Structured Light Scanner for Hyper Flexible Industrial Automation

Kent Hansen*†, Jeppe Pedersen*†, Thomas Sølund†‡, Henrik Aanæs‡, Dirk Kraft*
*The Maersk Mc-Kinney Moller Institute, University of Southern Denmark, Odense, Denmark
†Center for Robot Technology, Danish Technological Institute, Odense, Denmark
‡Department of Informatics and Mathematical Modelling, Technical University of Denmark, Copenhagen, Denmark
Email: thso@dti.dk (Thomas Sølund), aanes@dtu.dk (Henrik Aanæs), kraft@mmmi.sdu.dk (Dirk Kraft)

***Abstract*—A current trend in industrial automation implies a need for doing automatic scene understanding, from optical 3D sensors, which in turn imposes a need for a lightweight and reliable 3D optical sensor to be mounted on a collaborative robot *e.g.*, Universal Robot UR5 or Kuka LWR. Here, we empirically evaluate the feasibility of structured light scanners for this purpose, by presenting a system optimized for this task. The system incorporates several recent advances in structured light scanning, such as Large-Gap Gray encoding for dealing with defocusing, automatic creation of illumination masks for noise removal, as well as employing a multi exposure approach dealing with different surface reflectance properties. In addition to this, we investigate expanding the traditional structured light setup to using three cameras, instead of one or two. Also, a novel method for fusing multiple exposures and camera pairs is given.**

We present an in-depth evaluation, that lead us to conclude, that this setup performs well on tasks relevant for an industrial environment, where many metallic and other surfaces with difficult reflectance properties are in abundance. We demonstrate, that the added components contribute to the robustness of the system. Hereby, we demonstrate that structured light scanning is a technology well suited for hyper flexible industrial automation, by proposing an appropriate system.

***Index Terms*—3D reconstruction; structured light; robotics; 3D robot vision; large-gap gray code; data fusion**

I. Introduction

In the quest for increasing worker productivity, a trend in industrial automation is hyper flexibility, *i.e.*, that automation systems are able to adapt to different product types, without the intervention of highly skilled engineers. This will allow for a significant degree of automation in small and medium sized companies, where batch sizes are typically smaller, and thus automation systems need to be flexible, for them to be profitable [1]. In addition, the robots have to work without fences to enable cooperation between the robot and a human in object handling applications. This implies use of small robots not able to carry existing heavy 3D scanners as Gom Atos [2] and Steinbichler Comet [3]. To achieve hyper flexibility, scene understanding and versatile lightweight sensors suited for mounting on a collaborative robot are required in many cases, making vision sensors — especially 3D vision sensors — a central part of the sensor portfolio.

Accommodating this need for hyper flexibility implies several challenging issues from a computer vision perspective, mainly related to the complex geometries of the objects viewed, and the optical properties of these objects. Regarding the latter, metallic objects are very common in *e.g.*, bin-picking [4] and welding applications [5] in the manufacturing industry. In addition to this, for many applications the vision sensor should be mounted on a robot arm, to achieve the needed flexibility, constraining the weight and extent (*e.g.*, baseline) of the sensor.

In general structured light scanning is a common choice for accurate 3D surface reconstruction, and would as such be an ideal candidate for the task, *e.g.*, due to its relatively high speed and low cost. However, most such methods are only applicable for *well behaved scenes* that have limited interreflection, subsurface scattering and dynamic range [6]. In this paper, we present an aggregated structured light system, exploiting several recent advances within the field, as well as some practical innovations of our own. A main contribution of this paper is demonstrating, that this system can handle several characteristic situations related to industrial automation, thus implying, that structured light is a viable and versatile 3D sensor solution for hyper flexible industrial automation. Specifically, our evaluation shows that we are able to achieve an improvement over the basic implementation and some commercial products in all the relevant criteria we investigate.

The proposed system consists of three cameras and a projector designed for tool mounting on a small industrial robot arm. The system extends basic structured light by using (a) Large-Gap Gray codes to enhance the working range, (b) automatic creation of an illumination mask for noise removal, (c) three cameras for better coverage and specularity handling (a novelty in a robot mounted structured light system) and (d) multiple exposures to deal with different surface reflectance properties. Furthermore we present a novel method for fusion of data from multiple exposures and camera pairs.

This paper is structured with the following section discussing related work, Section III describing the hardware setup and giving on overview over the process flow, Section IV describing components of the reconstruction process, and Section V containing an extensive evaluation of the proposed system. The paper is concluded in Section VI.

II. Related Work

Structured light has received much attention in the four decades the technology has existed, and it is beyond the scope of this paper to give a full review of the field, which can

978-1-4799-7000-1/14 $31.00 © 2014 IEEE
DOI 10.1109/3DV.2014.53

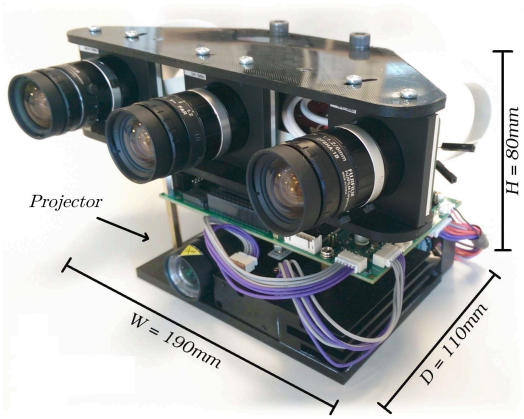


Fig. 1: The RoboVision3D scanner consisting of three cameras and a projector. The total weight of the system is 1030 grams.

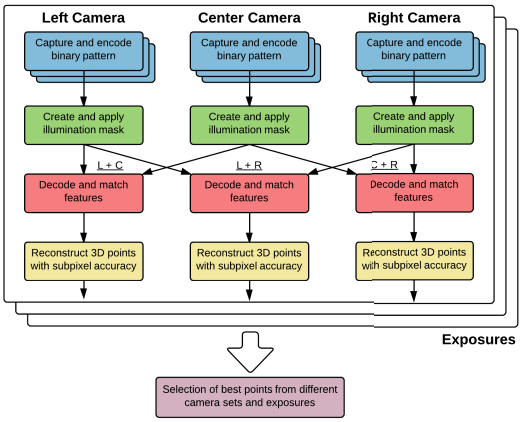


Fig. 2: Processing flow of 3D sensor based on structured light with three cameras and multiple exposures.

be found in [7]. Some typical applications for robot mounted structured light sensor systems are quality control [2], [3], bin-picking [4] and welding [5]. Recent advances in reconstruction methods include fringe projection [2], [3], LED grid pattern projection [4], projection of multiple planes [5], high speed line striping [8], wave grid patterns [9] and others. As *e.g.*, pointed out by Gupta *et al.* [6] structured light is sensitive to certain radiometric properties of the surfaces scanned such as inter-reflections, specularities, subsurface scattering and defocusing. Choosing binary patterns that ensure even light distribute as Goddyn *et al.* [10] and mentioned in [6] is a common way to reduce radiometric effects. Gupta *et al.* [6], sums up a body of work based on the discovery that many of the undesirable surface radiometric properties, are attenuated by high or low frequencies in the projected patterns. The conclusion is, thus, that one should adaptively chose between high or low frequency patterns, or use patterns with only middle frequencies.

Combining a digital micromirror device (DMD) laser projector which posses very high depth of field, with small aperture cameras can further increase the depth of field of structured light sensors as reported by [8].

Another challenge is that the limited dynamic range of the cameras is too constrained for robot picking applications, where objects with different material properties are common. A robust solution proposed in the literature, inspired by high dynamic range (HDR) imaging, is to do structured light scanning with different exposures (*i.e.*, exposure times) [11], [12]. Previously, two different approaches have been proposed for merging the differently exposed images to a single reconstruction, namely doing image fusion [12], *i.e.*, actually forming a HDR image, or doing point fusion [11], where the most reliable point estimate is chosen across the exposures. To the best of our knowledge, no previous comparison of these two approaches has been presented.

A contribution of this work is the use of three cameras instead of two in a system suitable for mounting on a robot arm. This non-fixed system has large advantages in relation to specularities, as well at resolving the occlusions related to parallel surfaces relevant for grasping. Previously multiple camera systems have been reported, *e.g.*, [9], [12], but all these were fixed systems, to the best of our knowledge, *i.e.*, not mountable on a robot arm. In regard to this trinocular approach, we also propose a new fusion method for merging the 3D reconstructions from the three camera pairs.

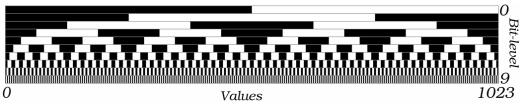
III. System Overview

Our proposed system, RoboVision3D, is based on a structured light approach and contains three cameras (cameras: VRmagic VRmS-16 1280x960, lenses: Fujinon DF6HA-1B 6mm C-mount) and a projector (TI DLP® LightCrafter™ 4500) as shown in Fig. 1. The system is designed for tool mounting on a small industrial robot arm and is thus small and lightweight. The two outermost cameras have a baseline of 15 cm and both outer cameras have a baseline of 7.5 cm to the center camera. The aperture of the cameras is set to f/8, and the aperture of the projector is fixed at f/1.2. To set our timings in perspective, all processing is performed on a laptop with 2nd generation i7 dual core processor and 8GB RAM.

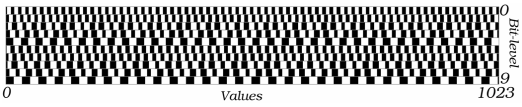
From a hardware point of view, it should be noted, that we employ three cameras in our setup, the benefits of which we describe in Section IV-D. The processing pipeline is illustrated in Fig. 2, and specific details hereof are presented in the next section.

IV. Components for Robust Reconstruction

This section presents the components of our structured light pipeline, that sets it aside from a traditional approach, and

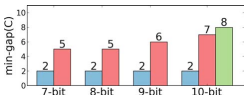


(a) Binary Reflected Gray Code (BRGC)

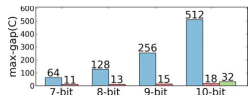


(b) Large-Gap Gray Code (LGGC)

Fig. 3: Comparison of 10-bit Gray codes.



(a) Minimum stripe width.



(b) Maximum stripe width.

Fig. 4: Comparison of stripe widths of BRGC (blue), LGGC (red) [10] and "Maximum min-SW" (green) [6] (The latter is only mathematically derived for 10-bit). To address illumination challenges patterns with small maximum width and large minimum width are desirable [6]. Here it is seen that the LGGC patterns, employed in our system, perform almost as well as the ones from [6], but are available at several projector resolutions.

provides the reported robustness in hyper flexible industrial applications.

A. Large-Gap Gray Code

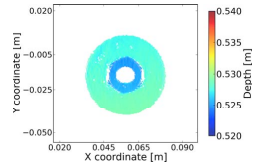
As mentioned, Gupta *et al*. [6] have pointed out that traditional binary reflected Gray codes (BRGC) have a suboptimal distribution in the frequency domain, and in turn proposed new binary stripe patterns. Instead of using the patterns proposed in [6], we use the Large-Gap Gray code (LGGC) of [10] inspired by the discussion in [6]. A visual comparison is shown in Fig. 3. This choice is motivated by a need for doing structured light at different projector resolutions, and as seen in Fig. 4 we achieve results similar to the proposal from [6] — *cf*., [13] for further details.

B. Illumination Mask from Separation of Light Components

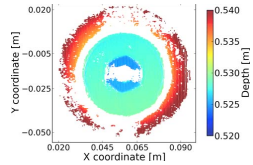
A necessary part of a structured light system is an illumination mask for each camera, which classifies each pixel as being illuminated by the projector or not. In our system we use the method by Nayar *et al*. in [14], with the exception that we use the LGGC patterns instead of the checkerboard pattern proposed in [14]. We also tried using the BRGC patterns, but achieved better results with the LGGC patterns.

The method of Nayar *et al*. [14], works by the idea that even though one pattern does not illuminate all possible parts of the scene, the combination of patterns does. Thus, the direct light component, L_d, can be computed via the maximum pixel

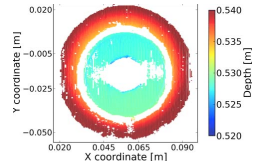
(a) Dark/bright suction cup.



(b) Low exposure: 2ms



(c) Medium exposure: 10ms



(d) High exposure: 40ms

Fig. 5: Illustration of the problem of scanning an object with multiple reflectances with a single exposure.

intensities over all patterns, and the global component via the minimum. Ambient light can be removed from the global component by acquisition of an additional image with the projector switched off, which makes it possible to obtain the indirect component, L_i, of the projector. These values can be used to create an illumination mask, M:

$$M(x,y) \quad = \quad \begin{cases} 1, \text{if } L_d(x,y) - L_i(x,y) \geq K_{noise} \\ 0, \text{otherwise.} \end{cases} \qquad (1)$$

for pixel x, y, where K_{noise} is a threshold related to the image noise level. For further details refer to [13].

C. Fusion of Multiple Exposures and Integration

A limitation of the basic approach to structured light scanning is that the inherent limitation in the dynamic range of the camera, as previously noted in *e.g*., [11], [12]. In these works the issue is addressed by using different camera exposures inspired by high dynamic range (HDR) imaging. As an example of the issue, consider Fig. 5 of a suction cup with a shiny metal bolt surrounded by dark rubber, making it impossible to do a 3D reconstruction with only one exposure.

In this work, we propose a new method for combining these multiple exposed images into a single 3D reconstruction. The idea is that, instead of explicitly forming a HDR image as done in [12], we specifically choose the most reliable edge correspondence over all possible exposures. That is, each 3D point is reconstructed from a pair of matching edges, and we have devised a method for determining which edge match gives the most certain 3D point. This is similar to the approach of [11], but with a different certainty measure suited for binary patterns. This certainty score is derived from the local per row image average and difference as illustrated in Fig. 6.

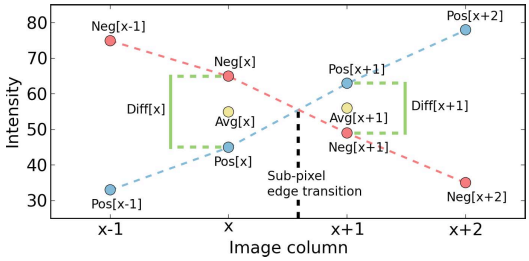


Fig. 6: Illustration of an edge transition in an image row. $\text{Pos}[x]$ and $\text{Neg}[x]$ denotes the captured positive/negative images of the pattern at column x. These values are used to compute the average values (yellow) and difference values (green) for certainty estimation of the edge.

$$\text{Avg}[x] = \frac{\text{Pos}[x] + \text{Neg}[x]}{2} \quad (2)$$

$$\text{Diff}[x] = \text{abs}\left(\text{Pos}[x] - \text{Neg}[x]\right) \quad (3)$$

We combine the following relevant measures: (the term I_{max} is used to denote the maximum global pixel intensity):

- **Average edge intensity (eq. 4)**
  This term can be used to select points from well-exposed areas and avoid points from edges being over- or under-exposed.

  $$\text{Edge}_{\text{avg}}[x] = \frac{I_{max} - \text{abs}(I_{max} - (\text{Avg}[x] + \text{Avg}[x+1]))}{I_{max}} \quad (4)$$

  The measure yields 1 for an average intensity of $\frac{I_{max}}{2}$, which is scaled towards zero for values getting near over- and underexposed intensities.
- **Edge contrast (eq. 5)**
  This measure can be used to select points with more perceptible edge transitions in the images.

  $$\text{Edge}_{\text{diff}}[x] = \frac{\text{Diff}[x] + \text{Diff}[x+1]}{2 \cdot I_{max}} \quad (5)$$

  The measure yields a high value for edges with a large difference across the edge.
- **Edge consistency (eq. 6)**
  This measure can be used to select points with more consistent average value across the edge, which indicates no changes in the underlying surface reflectance.

  $$\text{Edge}_{\Delta\text{avg}}[x] = \frac{I_{max} - \text{abs}(\text{Avg}[x] - \text{Avg}[x+1])}{I_{max}} \quad (6)$$

  The measure yields one for an equal average value, which is scaled towards zero for larger differences.

The three measures are combined into a certainty measure as follows:

$$\text{C}_{\text{Edge}}[x] = \frac{k_1 \cdot \text{Edge}_{\text{avg}}[x] + k_2 \cdot \text{Edge}_{\text{diff}}[x] + k_3 \cdot \text{Edge}_{\Delta\text{avg}}[x]}{k_1 + k_2 + k_3} \quad (7)$$

$$\text{C}_{\text{Point}} = \left(\frac{\text{C}_{\text{Edge}}[n] + \text{C}_{\text{Edge}}[m]}{2}\right) \cdot k_4 \quad (8)$$

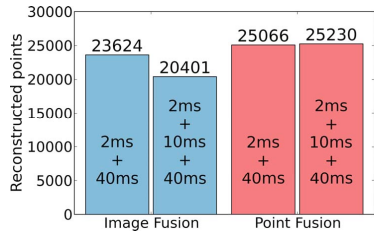


Fig. 7: Comparison of reconstructed points from HDR-like image fusion and point fusion using certainty estimation. Point fusion provides better results for both two and three exposures, while image fusion manages to lose information by addition of a third exposure.

The weight constants $k_1 \ldots k_3$ determine the weighting of the individual certainty factors described above. k_4 depends on which camera pair was used to reconstruct the given point, *cf.*, Section IV-D.

The method has been compared with HDR-like fusion of images with the projected pattern, which can be seen in Fig. 7. This result shows that more points are reconstructed using point fusion. The addition of a third exposure manages to eliminate points using image fusion, while it can only contribute with additional information using point fusion.

D. Three Camera Structured Light

An innovation in this work, is the use of three cameras in our robot mountable setup, as seen in Fig. 1. The motivation for doing this is two-fold, firstly it reduces the amount of occlusion in the 3D reconstructions, especially around parallel surfaces, which are of special importance for grasp planning, *cf.*, Fig. 8. Secondly, it drastically reduces the corruptive effects of specular surfaces, which are abundant with the many metallic objects in industrial applications. As seen in Fig. 2, we process the data from the three cameras, by computing a 3D reconstruction for each of the three camera pairs, and then merging them according to a reliability measure.

Concerning the specular surfaces, note that they cause large disruptive errors when a specular lobe or highlight is in the direction of the camera, *i.e.*, that if the surface were a mirror, then the camera could 'see' the light source. In the case of structured light, there is only one light source, *i.e.*, the projector, and as such there is in general only one lobe per specular surface point. In our three camera case, this observation, implies that any point on the surface can maximally be in the direction of one camera, so that all parts of the surface will be depicted highlight free in at least one camera pair. A practical implication of this is, that our system can work with significantly higher exposure times, without incurring large errors due to specularities.

The three depth maps from each of the three camera pairs are merged by transposing all matches into the reference frame of the left image and using the most reliable as determined by (8). To account for points from the different image pairs not

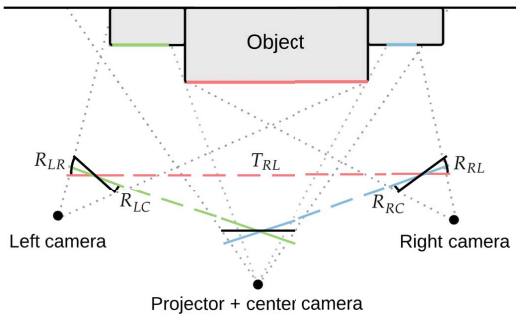


Fig. 8: Illustration of occlusion reduction by integration of a center camera. The black lines indicate the original image planes with corresponding optical center. The red part of the object is shared by all three cameras, while the green and blue part are only visible in the center camera and either left/right camera. The colored lines show the three rectified image planes, where the denoted translation and rotations are used to transform points into a common reference frame.

being perfectly co-located, the most certain over a small region is chosen. This gives good results, as seen in Section V.

V. Evaluation

Our RoboVision3D scanner will be evaluated based on the challenges in hyper flexible industrial automation, which can, based on the authors expertise in the field, be seen as:

- **Issue 1)** Requests for large working range, since the sensor placement can not always be chosen optimally.
- **Issue 2)** Ability to deal with a high diversity in objects from highly reflective metallic objects to extremely matte black objects.
- **Issue 3)** Ability to provide good coverage on objects with complex geometries.
- **Issue 4)** Resilience towards ambient light to avoid having to shield the automation setup.

Before going into the actual evaluation the general parameter setup will be discussed. Next the individual components in the method will be tested and finally the scanner will be compared to a list of similar 3D scanners using a mobile test plate and by scanning different surface reflectances.

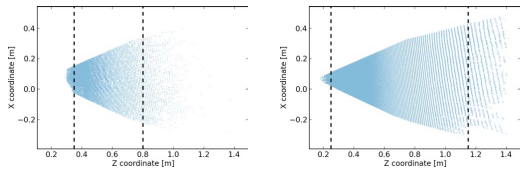
All evaluations are performed in typical factory lighting, which addresses *Issue 4* by working without having to shield the system.

A. General Setup and Parameters

For all experiments described in this section the sensor has been setup as described in section III. The system has a limited set of parameters, these have been determined as follows: The illumination noise rejection parameter K_{noise} has been chosen empirically to five for the 8-bit images used. The parameters $k_1 \dots k_4$ for the certainty weighting of the individual reconstructed points have been determined using a genetic algorithm (GA) optimization.

As input for the GA a scan of a plane at a known position was performed with multiple exposures. This made it possible to assign an error to each point and use the GA to optimize the coefficients in order to select the points with lowest errors based on the three measures and the camera pair used. The GA was run with a population of 25 with 100 generations and a mutation rate of 0.25.

The results from the GA optimization yielded weights of $k_1 = 0.4$, $k_2 = 0.99$, $k_3 = 0.36$ and $k_4 = 1$ if the left/right camera pair is used and $k_4 = 0.5$ if the center camera is combined with one of the others. This shows that the measure for average difference across the edge was found to be most important for point certainties and that the camera pair with the bigger baseline is considered to be more reliable.



(a) Depth map from BRGC (b) Depth map from LGGC

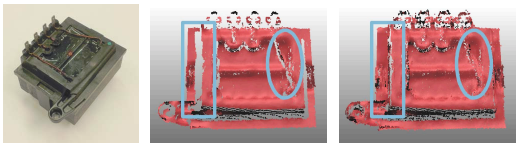
Fig. 9: Comparison of binary reflected Gray code and Large-Gap Gray code in terms of robustness towards projector defocus.

B. Test of Individual Components

The use of LGGC patterns was introduced to provide more robustness towards defocusing compared to conventional BRGC patterns. This has been verified by scanning along a flat surface with both patterns, which produces defocused stripes near and far away at the same time. The scanner was placed so the projector illuminated a depth from about 0.2–1.5 m. The results from this test is shown in Fig. 9. Here it can be seen that the BRGC pattern is able to reconstruct the surface from about 0.35–0.80 m, while the LGGC pattern reconstructs from about 0.25–1.15 m. This shows that LGGC is much more robust towards defocusing and thereby increases the working range of the 3D scanner addressing *Issue 1*.

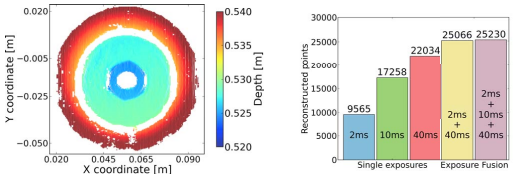
Another improvement has been made by integration of a third camera for better coverage on complex objects in relation to *Issue 3*. A result of this addition is shown in Fig. 10, where a transformer is scanned using two and three cameras respectively. The integration of the third camera yields about 30% extra points in this example, which shows the benefits in less occlusion and a more dense point cloud.

A method for fusing multiple exposures was developed, which addresses the challenges in *Issue 2*. A problematic object was shown in Fig. 5, where the results from multiple exposures can be seen in Fig. 11. This shows that the coverage is greatly improved and both the dark and bright areas are re-



(a) Transformer. (b) Two cameras. (c) Three cameras.

Fig. 10: Point clouds showing the additional coverage gained by adding the third camera. The highlighted areas show the added coverage clearly.



(a) Fusion result of three exposures. (b) Comparison of reconstructed points.

Fig. 11: Reconstruction results with exposure fusion of the suction cup from Fig. 5.

(a) Front (b) Back

Fig. 12: The mobile test plate from [15]. The test challenges 3D scanners in terms of reconstruction in textured areas, different surface properties and specular reflections from a metallic hemisphere. The test plate's size is approx. 30x30 cm and the hemisphere has a diameter of approx. 5 cm.

Fig. 13: The four additional hemispheres with various surface properties and the original hemisphere.

constructed simultaneously. The gain in terms of reconstructed points is approx. 15% in this example.

C. Evaluation using the Mobile Test Plate

A mobile test plate was proposed by Møller *et al.* in [15], where the performance of eight commercial 3D scanners was evaluated. The scanning distance was selected based on a bin-picking scenario, meaning that small robot-mountable sensors were tested at about 60 cm, whereas larger sensors were tested at a distance of about 1.5 m. We applied the same procedure to our RoboVision3D scanner, and furthermore evaluated a PrimeSense Carmine 1.09 RGB-D sensor for comparison.

The mobile test plate is shown in Fig. 12 with two sides that challenges the 3D scanners in different areas, which is mainly related to *Issue 2*. The front determines reconstruction of textured areas and different surface properties and the back evaluates reconstruction of a metallic hemisphere. The reader is referred to the original paper for further details about the design and construction of the test plate.

The quantitative results are listed in table I, which is an extension of the results from Møller *et al.* [15].

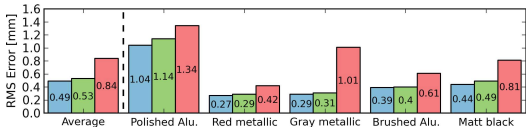
The 3D scanners are evaluated on several areas. The *RMS Error in Gray Region* is used as reference, where the RoboVision3D has an error of 0.1 mm, which is the lowest error among the scanners. The long range scanners from SICK and HDI provide results close to this error, however the only robot-mountable scanner with results near this is the SCAPE Grid Scanner. The *Max. RMS Error in Structured Region* evaluates the accuracy in textured areas, where the RoboVision3D scanner also provides the lowest error of 0.2 mm among the scanners. *Discretization* of the RoboVision3D scanner was measured as part of the evaluation procedure. The depth discretization was determined to 0 mm due to sub-pixel accuracy and the in-plane 1/2 discretization yields an equal resolution in the vertical and horizontal direction of 0.5mm, which is the finest resolution of the tested scanners. The RoboVision3D scanner was also evaluated based on reconstruction of a metallic hemisphere, where a *Fraction of Points on hemisphere* of 73% was obtained. This coverage is exceeded by some of the other 3D scanners, however the *RMS Error on Hemisphere Top* of 0.3 mm for RoboVision3D is significantly better than all the other scanners.

The test by Møller *et al.* [15] was extended with four additional hemisphere as shown in Fig. 13, which challenges the scanner further with darker/brighter reflectances in an angle towards the scanner. Fig. 14 shows the results for the RoboVision3D scanner with both single- and multi-exposure, where a low and high exposure are combined and from the PrimeSense Carmine 1.09 sensor.

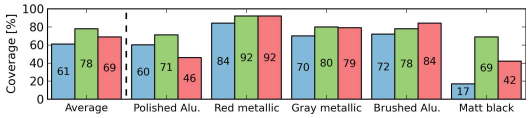
The average results show that RoboVision3D is more accurate compared to PrimeSense Carmine 1.09, however the coverage is 8% lower with single exposure and 9% higher with multi-exposure. The coverage is quite equal for the red metallic, gray metallic and aluminum hemispheres for the three sensors. The polished aluminum hemisphere is better reconstructed using RoboVision3D compared to PrimeSense Carmine 1.09 for both single- and multi-exposure. Furthermore it is worth to note that only 17% of the matt black hemisphere is reconstructed from RoboVision3D with a single exposure, however the multi-exposure configuration manages to reconstruct 69%.

Scanner @ 320 Lux Ambient Light	Front of Test Plate					Back of Test Plate	
	RMS Error gray region	Max. RMS Error structured region	Depth	Discretization In-plane 1	In-plane 2	Fraction of points on hemisphere	RMS error on hemisphere top
SICK Ranger 50E @ 2 m	0.5 mm	0.7 mm	0.3 mm	1.4 mm	2.8 mm	66%	0.9 mm
SICK Scanning Ruler @ 1.5 m	0.3 mm	0.6 mm	0 mm	2.3 mm	2.4 mm	78%	1.2 mm
HDI Advance R2 @ 2.2 m	0.3 mm	0.3 mm	0 mm	0.6 mm	0.6 mm	5%	N/A
SCAPE Grid Scanner @ 60 cm	0.4 mm	0.7 mm	0 mm	5.0 mm	5.2 mm	21%	1.3 mm
PMD CamCube 3.0 @ 60 cm	9.1 mm	13 mm	0 mm	1.6 mm	1.9 mm	100%	12 mm
Fotonic C70 @ 90 cm	1.4 mm	1.5 mm	0.5 mm	6.0 mm	7.6 mm	65%	13 mm
Xbox Kinect @ 60 cm	1.2 mm	1.2 mm	2 mm	1.4 mm	1.4 mm	58%	1.7 mm
ASUS Xtion PRO @ 60 cm	1.3 mm	1.1 mm	2 mm	1.4 mm	1.4 mm	90%	1.5 mm
RoboVision3D @ 70 cm	0.1 mm	0.2 mm	0 mm	0.5 mm	0.5 mm	73%	0.3 mm
PrimeSense Carmine 1.09 @ 60 cm	0.7 mm	0.6 mm	1.0 mm	1.0 mm	1.0 mm	83%	1.0 mm

TABLE I: Evaluation of mobile test plate using results from [15, table 1], which are extended by results from the sensor developed in this work and PrimeSense Carmine 1.09 (highlighted in gray).



(a) RMS error on hemispheres



(b) Coverage on hemispheres

Fig. 14: Results from scanning the additional spheres with RoboVision3D single exposure (blue), multi exposure (green) and PrimeSense Carmine 1.09 (red).

Parameter	Scan distance Min 25cm	Center 70cm	Max 115cm	Unit
Physical size		190x80x110		(WxHxD) mm
Weight		1030		g
Image acquisition time		3052		ms
Post processing time		2846		ms
Total time		5898		ms
Scanning area	140x140	480x310	750x460	(WxH) mm
Horizontal resolution	0.24	0.51	0.85	mm
Vertical resolution	0.22	0.47	0.79	mm

TABLE II: List of specifications of the constructed vision platform.

D. Sensor Quantification

In table II the specifications of the developed 3D sensor are listed. It should be noted that the image acquisition time is rather high at three seconds, which is caused by the fact that our cameras are only able to run at 10 Hz because of the interface board used. Higher camera frame-rates could bring this time down significantly. With 30 Hz cameras the acquisition time would be closer to one second. Going beyond 30 Hz the system does not scale anymore since the processing time becomes the limiting factor.

VI. Conclusion

In summary, this paper presents a compact structured light sensor suited for robot tool mounting, and addresses the issues relating to hyper flexible automation well. It exploits the properties of LGGC code to improve the robustness towards defocusing, inter-reflections and subsurface scattering. The method enhances the depth of field from 0.45–0.9 m. The use of three cameras enables reconstruction of additional 30% points near abrupt depth discontinuities. Scenes with high dynamic range are handled by applying a novel algorithm for fusing the point estimates from images with different exposure,resulting in an additional 15% correct reconstructed points. The sensor is compared to the depth sensor evaluation in [15]. The presented sensor outperforms all sensors in [15] in terms of accuracy on a metallic hemisphere and still achieves good results for the fraction of correct points.

As future work, the acquisition time needs to be reduced by introducing cameras with higher frame rate. Methods for automatic selection of camera exposure and projector illumination needs to be implemented to remove the need for parameter adjustment.

Acknowledgements

The research leading to these results has been funded in part by the Danish Ministry of Science, Innovation and Higher Education under grant agreement #11-117524, #3067-00001B (MADE: Manufacturing Academy of DEnmark) and from the European Union's FP7 (FP7/2007-2013) work program under grant agreement #285380 (PRACE: The PRoductive robot ApprentiCE). The authors would like to thank Bent Møller for providing a measurement target and support in applying the test from [15].

References

[1] euRobotics aisbl, "Strategic research agenda," 2014. [Online]. Available: http://www.eu-robotics.net/cms/upload/PDF/SRA2020_0v42b_Printable_.pdf

[2] GOM Optical Measuring Techniques, "GOM ATOS Scanbox," 2014. [Online]. Available: http://www.gom.com/home.html

[3] Steinbichler Optotechnik GmbH, "Steinbichler COMET Automated," 2014. [Online]. Available: http://www.steinbichler.co.uk/products/surface-scanning/3d-digitizing/comet-automated.html
[4] S. Baby, K. B. Simonsen, I. Balslev, V. Kruger, and R. D. Eriksen, "3D Scanning of Object Surfaces Using Structured Light and a Single Camera Image," in *IEEE International Conference on Automation Science and Engineering*, 2011, pp. 151–156.
[5] M. Rodrigues, M. Kormann, C. Schuhler, and P. Tomek, "An intelligent real time 3D vision system for robotic welding tasks," in *9th International Symposium on Mechatronics and its Applications (ISMA)*, 2013, pp. 1–6.
[6] M. Gupta, A. Agrawal, A. Veeraraghavan, and S. G. Narasimhan, "A Practical Approach to 3D Scanning in the Presence of Interreflections, Subsurface Scattering and Defocus," *International Journal of Computer Vision*, vol. 102, no. 1-3, pp. 33–55, 2012.
[7] J. Salvi, S. Fernandez, T. Pribanic, and X. Llado, "A state of the art in structured light patterns for surface profilometry," *Pattern Recognition*, vol. 43, no. 8, pp. 2666–2680, 2010.
[8] C. Mertz, S. J. Koppal, S. Sia, and S. G. Narasimhan, "A low-power structured light sensor for outdoor scene reconstruction and dominant material identification." in *CVPR Workshops*, 2012, pp. 15–22.
[9] N. Kasuya, R. Sagawa, H. Kawasaki, and R. Furukawa, "Robust and accurate one-shot 3d reconstruction by 2c1p system with wave grid pattern," in *Proceedings of the 2013 International Conference on 3D Vision*, 2013, pp. 247–254.
[10] L. Goddyn, G. M. Lawrence, and E. Nemeth, "Gray Codes with Optimized Run Lengths," *Utilitas Mathematica*, vol. 34, pp. 179–192, 1988.
[11] D. Scharstein and R. Szeliski, "High-Accuracy Stereo Depth Maps Using Structured Light," in *IEEE Computer Vision and Pattern Recognition*, 2003, pp. 195–202.
[12] M. Weinmann, C. Schwartz, R. Ruiters, and R. Klein, "A Multi-camera, Multi-projector Super-Resolution Framework for Structured Light," *2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pp. 397–404, 2011.
[13] J. Pedersen and K. Hansen, "Robot Mountable Embedded Vision Platform for Multimodal Point Cloud Acquisition in Industrial Applications," Master's thesis, University of Southern Denmark, 2014.
[14] S. K. Nayar, G. Krishnan, M. D. Grossberg, and R. Raskar, "Fast separation of direct and global components of a scene using high frequency illumination," *ACM Transactions on Graphics*, vol. 25, no. 3, p. 935, 2006.
[15] B. Møller, I. Balslev, and N. Krüger, "An Automatic Evaluation Procedure for 3-D Scanners in Robotics Applications," *IEEE Sensor Journal*, vol. 13, no. 2, pp. 870–878, 2013.

3.7 Discussion and Conclusion

In this section the general aspects of estimating the topology of objects with optical methods are presented. Pros and cons of the different methods have been discussed in relation to the requirements of industrial automation and robotics. A new structured light scanner suited for tool mounting on a collaborative robot is proposed, which combines novel structured light techniques in a small physical unit. The sensor solves some of the challenges that exist with conventional 3D sensors targeting industrial automation, which are robustness towards reflection, a high dynamic range and large depth of field. If inexpensive small commercial 3D sensor systems have to be deployed in industrial productions in the future it is important that many of the novel research results in structured light methods are integrated e.g. robustness towards, inter-reflections, sub-surface scattering and projector defocus.

Today, the marked for 3D sensors is dominated by sensors for consumer electronics where the development within 3D estimation techniques is going fast. In the future this will effect the development of industrial 3D sensors, which is mainly dominated by laser line scanners today. Structured light scanning technologies are still dedicated high end metrology applications. In this domain the structured light scanners feature all the newest state-of-the-art technologies e.g. fringe scanning, High Dynamic Range and methods for reducing inter-reflections. When the fast development which is characterizing consumer electronic development enters the marked for industrial 3D sensors, hopefully we will see low cost 3D picking system for robots on the marked. The research in the field is mature but the first low cost products, which are capable of reconstruction objects with reflective surfaces are missing. Leading sensor manufactures for the automation industry like SICK and Leutze are not offering any structured light device despite the proved quality. However, new companies like ShapeCrafter have seen the marked potential and introduced a structured light sensor targeting the automation industry.

Chapter 4

3D Pose estimation

4.1 Introduction

The ability to recognize and compute object poses in range images has been an interesting research problem for decades. However, the benefits going from 2D image domain to recognition in the 3D domain have become more and more evident and applicable during the last ten years. The main reasons are that computers today have more computational power to handle 3D data and that 3D sensors are becoming better and cheaper. With the introduction of the Microsoft Kinect sensor in 2010 the amount of research and industrial applications using 3D computer- and robot vision have increased dramatical. Applications such as perception for autonomous driving, reverse engineering, robot guidance, robot bin-picking, visual odometry for mobile robot navigation, 3D modelling and 3D tracking all benefited from cheaper 3D sensors and increasing computational power. Especially, the robotic domain befits from 3D data of the simple reason that robots operate in a 3D world. Contrary to 2D robot vision the third dimension is explicit given by applying 3D robot vision techniques. Furthermore, a step into the 3D world enables robots to reason about geometry and make use of techniques and algorithms defined in the computationally geometry domain to compute poses of objects. This chapter will focus on general advances and state of the art within 3D pose estimation and local shape features.

The data representation for any 3D pose estimation algorithm is at the lowest level a collection of observed point in a 3D euclidean space. Data is typically captured by a 3D scanner device and represented as a 2.5D range image or a 3D point cloud. For some sensor types e.g. RGB-D sensors and some structured light scanners, each point in the point cloud has a 6D coordinate which besides an x,y,z coordinate, includes RGB color information. From this data representation, the objective of pose estimation is to estimate the (rigid) Transformation $\boldsymbol{T} \in \boldsymbol{SE(3)}$ that minimizes the mean square distances between each point in an object model $\boldsymbol{M_q}$ and the corresponding point in a scene $\boldsymbol{S}$. A prerequisite for successfully estimating the pose of an object in a scene is that the scene representation is fairly complete, which requires the right 3D sensor technology to ensure that an objects fine structure and curvature are captured with limited noise. This topic was covered in Chapter 3.

The mathematical definition of pose is different depending on the problem domain. In this thesis a pose is defined as the transformation $\mathbf{T}$ in euclidean space $\boldsymbol{SE(3)}$ that aligns the object mode $\mathbf{M}$ with the model present in the scene $\mathbf{S}$. The definition of pose is given by the homogeneous transformation matrix in Equation 4.1. This transformation is sometimes referred to as the rigid body motion.

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{R_{3x3}} & \boldsymbol{t_{3x1}} \\ \boldsymbol{0_{1x3}} & \mathbf{1} \end{bmatrix}; \quad \boldsymbol{R} \in \boldsymbol{SO(3)},\ \boldsymbol{t} \in \mathbb{R}^{\mathbf{3}} \tag{4.1}$$

In this chapter, the fundamental steps for estimation the pose of objects in a 3D scene is covered. In Section 4.2 a short description of the current state-of-the-art in commercial 3D picking systems is presented. The focus of this chapter is state-of-the-art local shape features, which are presented in Section 4.3. In 3D pose estimation, description of local shape features is the key to successfully detect objects. The matching and estimation step is similar to the pipeline in Section 2 and is therefore left out in this chapter. The contributions within 3D pose estimation and model learning for 3D pose estimation are presented in Section 4.4. [Contribution E] deals with a vision system for fast teaching of object models for 6D pose estimation and [Contribution F] presents a novel large-scale dataset and a benchmark of current state-of-the-art local shape features. This chapter is summarized in Section 4.5.

4.2 Commercial products for 3D picking

Three-dimensional perception and imaging is a fundamental technology for robots in the future. The ability to compute six degrees of freedom poses enables robots to react to the world around them. However, only very few commercial machine vision software tools exist on the market to process 3D data and compute object poses from 3D range images. The most common software tools for 3D data processing are still metrology software packages as described in Section 3.2. Machine vision software libraries like Halcon [1], Matrox [2], Scorpion [3], Candelor [4], Aqsense [5] and Common Vision Blox [6] all include basic methods for 3D processing and pose estimation. However, the functionalities in e.g. Halcon is limited and the pose estimation algorithm included is a algorithm from 2010 [DUNI10], which probably is optimized since then. The reason for the limited number of tools could be that 3D pose estimation is still today in its early stage and the marked pull is still limited because of high prices. The tendency is the same for commercial 3D Picking systems for robot guidance. Only a limited amount of products are available. The products are mainly bin picking products because of the obvious advantages in applying 3D vision in this applications. The first commercial bin picking system on the marked is developed by the Danish company, Scape Technologies [7]. Since they introduced an industrial bin picking system around 10 years ago, several companies have released bin picking products. However, the number of solutions are limited and many of the bin picking products are introduced on the marked during the last 5 years. This includes products like the Sick PLB 500 [8], Fanuc iRVision 3DL [9] and ISRA Vision ShapeScan3D [10].

A new class of inexpensive 3D picking systems have started to reach the marked. The functionality of these systems are limited and are not as comprehensive as the full bin picking products. However, the 3D picking systems solve limited number of problems. An example of these low cost systems is the PickIt3D

[1] MvTec Halcon - `http://www.halcon.com/`
[2] Matrox - `http://www.matrox.com/imaging`
[3] Scorpion Stinger - `https://scorpion3dstinger.com/`
[4] Candelor - `http://candelor.com/`
[5] `http://www.aqsense.com`
[6] Common Vision Blox - `http://commonvisionblox.com`
[7] Scape Technologies - `http://www.scapetechnologies.com`
[8] Sick PLB - `https://www.sick.com/dk/en/system-solutions/robot-guidance-systems/plb/plb-500/p/p294546`
[9] Fanuc iRVision 3DL - `http://robot.fanucamerica.com/products/vision-software/robot-vision-software.aspx`
[10] ISRA Vision ShapeScan3D - `http://www.isravision.com/en/robot-vision/shapescan3d`

Figure 4.1: **Upperleft:** Scape Technologies binpicker with Scape 3D Grid scanner **Upperright:** ISRA Vision ShapeScan3D. **Midleft:** Fanuc IrVision 3DL. **Midright:** SICK PLB 500 bin picker. **Bottom:** Intermodalic PickIt3D.

product from Intermodalic [11], which is a system which is able to pick simple shapes from boxes and bins by using a Kinect like sensor. The first version of the system is limited to handle objects, which the kinect sensor is able to reconstruct and not capable of estimating the pose of advanced geometries using a CAD model.

During this industrial Ph.D project several of these systems have been explored. The experience from tests and surveys have shown that these systems struggles with the same challenges as stated in this thesis. They all have problems in reconstruction reflective surfaces and scenes with large amount of inter-reflection and sub-surface scattering. Furthermore, many of the systems are not capable of detecting simple objects without many distinct 3D shape features. This

[11] PickIt3D - `http://www.pickit3d.com/en`

problem is empirical proven in [Contribution F] later in this chapter.

4.3 Local Features

Local feature descriptors are a natural part of any 3D object detection pipeline. The objective in local shape feature description is to compute a simpler, denser, complete and distinctive representation of a 3D surface. The main properties of a good 3D local feature descriptor is that it needs to be invariant to rotation and robust to noise, varying mesh resolution, occlusion, clutter and other nuisances. Local shape features is particlar usefull in for 3D pose estimation in presence of occlusion and clutter. In this section, local shape features are covered. Global shape feature description techniqes where one full model signature is computed as known from e.g. from shape retrival algorithms [TV04] is not a part of this chapter.

4.3.1 Local feature descriptors

Local shape descriptors capture useful statistic of the geometry in a local region around a feature point. In the 2D domain, local features capture the local appearance around a point as described in Section 2.4. This appearance in the image domain is directly effected by the surrounding illumination, which can cause similar objects to appear very differently under different illumination condition and view points. In 3D applications where points are represented without appearance information like color or intensity, this issue disappears, which makes the feature description more stable to out-coming phenomenons. It is clear that the challenges with ambient illumination do not disappear but the problem is now a part of the sensor technology as discussed in Chapter 3. This gives a clear division of the different problems. Many of the challenges that exist in 2D are today solved with dedicated algorithms and external light settings but it is special settings and not a part of a general sensor. The advantage of 3D geometry compared to 2D appearance is that the geometry of an object is similar from different view points. This is not always the case in 2D images where pixel intensities can vary from different views. However, a 3D detection pipeline needs to be able to handle missing points while 2D detection pipelines have to handle wrong pixel measurements. One common error source which always is present, is noise.

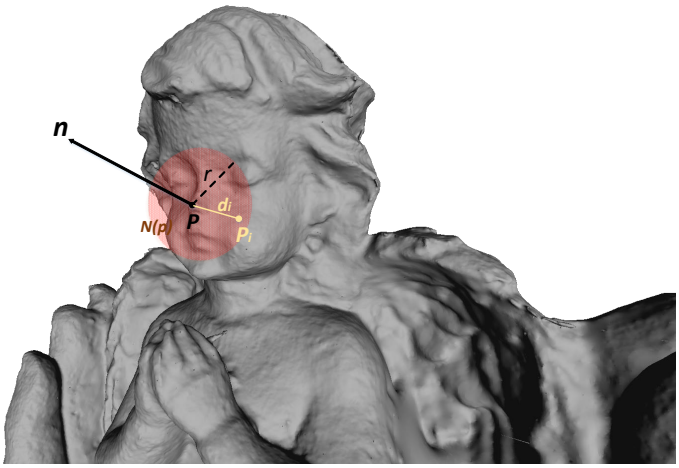


Figure 4.2: The basic elements of local shape description are a normal vector $\hat{\boldsymbol{n}}$, a support region $\boldsymbol{n(p)}$, a support radius $\boldsymbol{r}$, a keypoint $\boldsymbol{p}$ and point relations e.g. $\boldsymbol{(P_i, d_i)}$

During, feature description the underlying 3D shape of a model $\boldsymbol{M_q}$ is considered and a feature vector is computed for each keypoint. Typically, the feature description is aligned with the surface normal $\boldsymbol{n}$ or a Local Reference Frame (LRF) is constructed. Local Reference Frames are covered in Section 4.3.1. A visualization of the general description process is shown in Figure 4.2. Here a surface point $\boldsymbol{p}$ is described by considering the neighbor points $\boldsymbol{N(p)}$ within the support radius $\boldsymbol{r}$. The support radius containing all considered neighbouring points and is outlined by a red sphere. A feature vector is now computed by considering the spatial distribution of points or computing geometric relations between points in the neighbourhood. Local features are catagorized into three main groups; signature, spatial histogram and geometric attribute histograms methods. This catagorization is adapted from [GBS+14]. State-of-the-art features which are based on the spatial distribution are reviewed in Section 4.3.2 and features based on geometric relations are reviewed in Section 4.3.3. Features computed as signatures are not reviewed but a comprehensive review is found in [GBS+14]. Only few of the proposed local feature descriptors make use of radiometric properties around the point like color or intensity. In the section a description of the local features proposed in literature is reviewed and categorized.

Much research on discriminative local shape descriptors exist. Recently, a study

by Buch et.al. [BPK16] and Guo *et al.* [GBS+16] pointed out that state of the art local shape feature descriptors are not adequate in the description of the shape. The study found that the best feature descriptor in one test dataset was not necessarily the best feature descriptor in another dataset. This implies that more discriminative power is needed in order to solve the pose estimation problem with one dedicated shape feature. This is especially of interest in robotics where algorithms with as few tuning parameters are wanted. If a general Plug-n-Play 3D robot guidance vision system has to be realised, one general feature or a set of features, which are complementary is required. With such a system, only a CAD model of the object is needed to make the system detect object instances. Furthermore, the study from Buch et.al. [BPK16] exactly showed that the combination of already existing feature descriptors together with a simple feature selection criterion, based on the L2 distance in feature space, are increasing the performance over many datasets. This result shows that instead of aiming at finding the best general feature descriptor, an algorithm that combines existing feature descriptors in a coherent framework with an intelligent or learning based feature selection method, could be the solution to the pose estimation problem in the search for one general algorithm.

As the study from Buch *et al.* [BPK16] and Guo *et al.* [GBS+16] demonstrated, there is a remarkable difference in performance depending on which dataset is used in experimental evaluation of each local shape feature. This result implies that the research community has a general evaluation problem because of the sizes of the state-of-the art datasets. In [Contribution F] in Section 4.4 a new large scale dataset is proposed to provide the data foundation for better evaluation of 3D pose estimation algorithms.

Local features descriptors based on histograms capture the local neighbourhood around a keypoint by accumulating geometric or topological measurements e.g.(number of points, mesh area). Histogram-based-methods are the most common method for 3D local surface descriptors. One important property of local 3D features is invariance to rigid transformations, which for most state of the art features, are ensured by creating a Local Reference Frame(LRF). Local reference frames are established to make the feature description relative to the reference frame and not to a given view point. A local reference frame could be established utilizing the normal vector of the keypoint as the z-axis and e.g. the direction of the principal component of the data as the x-axis. Then the y-axis of the LRF is easily computed by the dot product of the x/z axis. This approach was first proposed by Taati *et al.* [TBJG07] who obtained an LRF by Principal Component Analysis of the covariance matrix of the point set included in the support region. During the last decade the 3D object recogniton research community has proven that construction of a robust and reliable LRFs are essential in order to get good recognition performance. The repeatability of

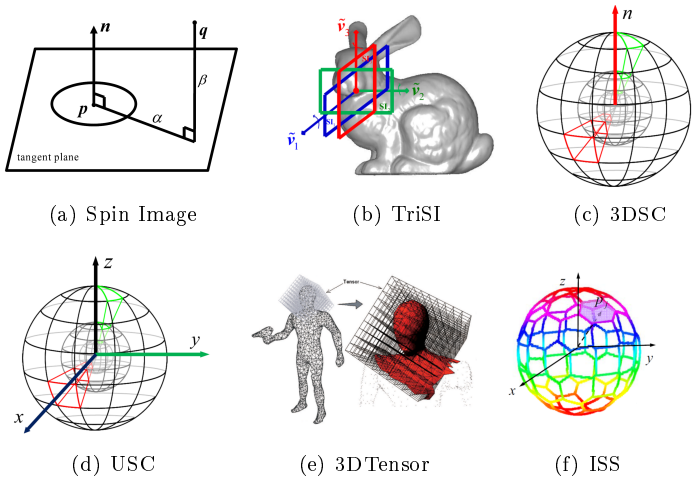
an estimated LRF directly affects the robustness and descriptiveness of local 3D feature descriptors. Local feature descriptors with a low repeatable LRF will result in poor matching performance [PS11].

The types of histograms which state of the art local features are applying can further be divided into three different sub-groups [GBS+14]; spatial distributed histograms, geometric attribute histograms and oriented gradient histograms. In the following state of the art methods using spatial distributed histograms and geometric attribute histograms are presented.

4.3.2 Spatial distributed Histograms

Local features that generate spatial distributed histograms, establish a descriptor of the neighborhood around a keypoint according to the spatial distribution of the points within the support radius e.g. the point coordinates. Johnson and Herbert [JH99] where some of the early pioneers in 3D local surface description. Their spin image feature first creates a local reference axis from the keypoint $\boldsymbol{p}$ normal vector $\boldsymbol{n_p}$ and express neighbouring points by two parameters; the radial distance $\boldsymbol{\alpha}$ and signed distance $\boldsymbol{\beta}$, see Figure 4.3(a). $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ forms a cylindrical coordinates of a neighbouring point $\boldsymbol{p_n}$ where $\boldsymbol{\alpha}$ is computed as the radial distance that lies on the tangential plane to the point $\boldsymbol{p}$, $\boldsymbol{\alpha = \sqrt{||q-p||^2 - (n \cdot (q-p))^2}}$ and the signed distance $\boldsymbol{\beta = n \cdot (q-p)}$. The $\boldsymbol{(\alpha, \beta)}$ space is then projected into a single 2D "image" for each keypoint $\boldsymbol{p}$ by accumulating $\boldsymbol{(\alpha, \beta)}$ coordinates into bins and each bin is bilinearly interpolated. Spin images are robust to clutter, occlusion and rigid transformations but are sensitive to the mesh resolution and non-uniform sampling [MBO10]. Today, spin images are considered as the de facto baseline feature in benchmarking of 3D local feature descriptors [TSDS10],[GSB+13b], [GBS+16]. Varieties of Spin images have been proposed like spin image signatures [ABBP07], a multi-resolution spin image [DK06], a spherical spin image [RCSM01], a scale invariant spin image [DK12] and a color spin image [PZC13]. Recently, the Tri-Spin-Image (TriSI)[GSB+15] descriptor was proposed. It generates three spin images from the coordinate axis of the LRF for each keypoint detected in a mesh, see Figure 4.3(b). The three spin images are the concatenated and compressed by projecting the TriSI feature to a PCA subspace. In a recent evaluation of features [GBS+16], the TriSi feature showed the best scalability with respect to the number of models in a dataset.

The 3D Shape Context (3DSC) [FHK+04] is an extension of 2D shape context [BMP02]. The support region of 3DSC is a sphere centered at the keypoint $\boldsymbol{p}$



(a) Spin Image (b) TriSI (c) 3DSC

(d) USC (e) 3DTensor (f) ISS

Figure 4.3: Spatial distributed Histograms

with the north pole aligned with the normal $\boldsymbol{n_p}$. The sphere is divided into equal distributed bins along the azimuth and elevation dimension and logarithmic distribution in the radial direction, see Figure 4.3(c). The logarithmic distribution in the radial direction makes the descriptor more robust to shape distortion [FHK+04]. It is reported that 3DSC achieve higher recognition rate in noisy scenes compared to spin image. Sukno *et al.* [SWW13] resolved the asymmetry ambiguity of 3DSC, by adding a simple measure of rotational symmetry. Tombari *et al.* extended the 3DSC by associating each keypoint $\boldsymbol{p}$ with a repeatable and unambiguous LRF, computed by EigenVector Decomposition of the covariance matrix $\boldsymbol{M}$, see Figure 4.3(d). Experimental results showed that Unique Shape Context (USC) improved the accuracy of feature matching with less memory requirements compared to 3DSC. Mian *et al.* [MBO06] presented the 3D Tensor that constructs a 3D descriptor by choosing a pair of vertices which satisfy a distance and angle constrain to construct a LRF. A local 3D grid is then constructed and the surface area in each bin is summed see Figure 4.3(e). The 3D Tensor descriptor is robust to noise, occlusion and varying mesh resolutions. Experimental results showed that the 3D Tensor outperformed spin image in recognition rate. The Intrinsic shape signatur (ISS) [Zho09] defines a LRF for a keypoint by computing the eigen values $(\boldsymbol{e_1},\boldsymbol{e_2})$of the weighted covariance matrix of the neighbouring points, see Figure 4.3(f). Each neighbouring point is weighted to compensate for uneven spatial sampling. The (x, y, z) axis of the LRF is defined as $(\boldsymbol{e_1},\boldsymbol{e_2},(\boldsymbol{e_1} \cdot \boldsymbol{e_2}))$. Experimental results showed that ISS outperforms both spin image and 3DSC in term of the amount of correct feature matches in noisy, cluttered and occluded scenes. Guo *et al.* [GSB+13b]

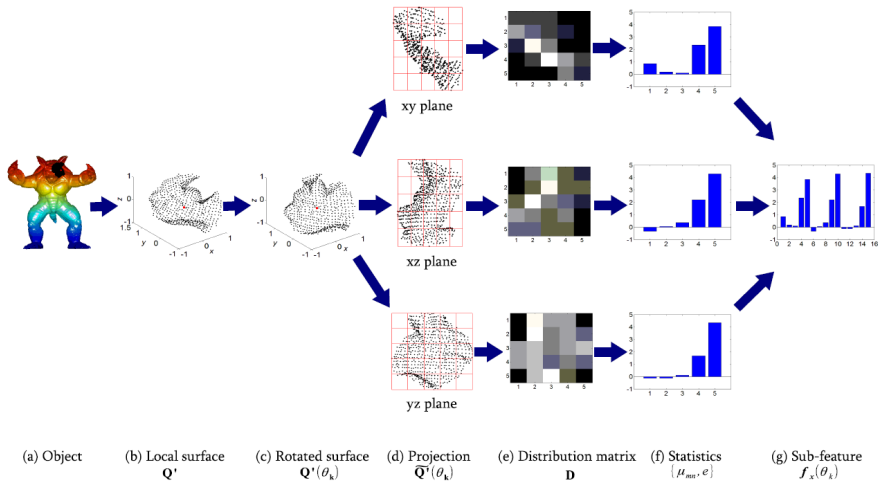


Figure 4.4: The Rops feature description [GSB+13b]

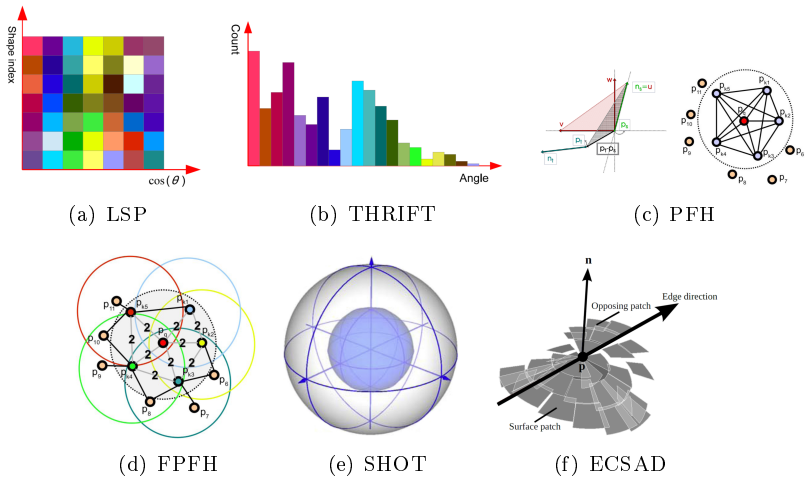
introduced the Rotational Projection Statistic (Rops) descriptor. Rops is the first descriptor to include the underlying local mesh surface and not only rely on the mesh vertex points during LRF construction. This methodology has been proven efficient in creating stable LRF which is more robust to noise and varying mesh resolutions. The LRF for the Rops descriptor is computed with an eigenvalue decomposition on the overall scatter matrix $\boldsymbol{C}$, where $\boldsymbol{C}$ is constructed from points lying at the $\boldsymbol{i}$th triangle mesh on the surface included in the feature support radii. The LRF is then defined as the three eigenvectors where the sign of each eigenvector is determined from the sign of the inner product of the eigenvector and the scatter vectors to avoid a sign ambiguity. The local point set $\boldsymbol{Q}$ around a keypoint, defined by the support radius, is aligned with the LRF Figure 4.4(a-b). $\boldsymbol{Q}$ is then rotated and projected into the $\boldsymbol{xy}$, $\boldsymbol{xz}$, $\boldsymbol{yz}$ planes and a rectangular bin of size $\boldsymbol{L}$ x $\boldsymbol{L}$ is created, see (Figure 4.4(d-e)). The number of points falling into each bin are summed and the bin is normalized to increase the robustness towards mesh resolution. This distribution matrix $\boldsymbol{D}$ is condensed by computing the central moment and the Shannon entropy and create three statistical vectors to increase the computational efficiency, see Figure 4.4(f). The three statistical features are concatenated into one sub-feature, see Figure 4.4(g). A set of sub features is created by rotating $\grave{\boldsymbol{Q}}$ around the LRF $\boldsymbol{x}$,$\boldsymbol{y}$,$\boldsymbol{z}$ and concatenating the computed sub-features into one feature descriptor. The Rops feature is considered as the state-of-the art local feature. Experimental results have shown that Rops outperforms Spin images [JH99], LSP [CB04], THRIFT [FDvdH07], SHOT [STS14] and MeshHOG [ZBVH09] in

presence of noise, varying mesh resolution and mesh holes. These experiments are based on data from up to four smaller datasets. However, a recent study by Buch *et al.* [BPK16] showed that the ECSAD [JBK15] and NDHIST [BPK16] features have equivalent matching performance on some datasets and objects. In [GSB+13a] the Rops features were extended to a color version C-Rops and a version that fuses both color and shape. The experimental results showed that combination of color and shape features increased the performance for objects with few geometrical features. However, applying Color-only-Rops (C-Rops) on geometrical rich objects results in poor performance compared to applying shape only (S-Rops) features.

4.3.3 Geometric Attribute histogram

Unlike, the feature descriptors presented above, geometric attribute histograms compute a feature around a local point by ensemble geometric relation using e.g. normals or curvatures of the local surface. One simple relation was proposed by Yamany and Farag [YF02], who accumulated simplex angles of the underlying mesh into a 2D histogram. For each keypoint $\boldsymbol{p}$ the distance to the neighboring points are represented in the first histogram dimension and the angle $\mathbf{arccos}\,\frac{\boldsymbol{n}\cdot(\boldsymbol{q}-\boldsymbol{p})}{||\boldsymbol{q}-\boldsymbol{p}||}$ in the second dimension. Similar to "Surface Signature" proposed in [YF02], the "Local Surface Patch" (LSP) [CB04] computes a 2D histogram containing the local surface shape index value vs. dot product of the surface normals between a keypoint $\boldsymbol{p}$ and its neighbours,see Figure 4.5(a). It was reported that LSP is as efficient for 3D object recognition than spin image but computational costly. The "THRIFT" descriptor propose in [FDvdH07] computes a 1D histogram of normal differences of a point $\boldsymbol{p}$, see Figure 4.5(b). The two normals are calculated by fitting two planes with a different window size to the local surface patch around the point $\boldsymbol{p}$ and compute the normals of the to virtual planes. Later, Flint *et al.* [FDvdH08] extended "THRIFT" to a weighted histogram containing the angle difference between the keypoint $\boldsymbol{p}$ normal and the neighbouring point normals $\hat{\boldsymbol{p}}$.

The "Point Feature Histogram" (PFH) introduced by Rusu *et al.* [RBMB08] encodes the local surface in the support region by randomly pairing all points in sets of two. As a first step, a Darboux frame is defined for each point pair $\boldsymbol{p_s}$ and $\boldsymbol{p_t}$ using the surface normal $\boldsymbol{u} = \boldsymbol{n_s}$ of the source point as x-axis, the vector $\boldsymbol{v} = (\hat{\boldsymbol{p}}_{\boldsymbol{t}} - \hat{\boldsymbol{p}}_{\boldsymbol{s}}) \times \hat{\boldsymbol{n}}_{\boldsymbol{s}}$ as y-axis and the vector $\boldsymbol{w} = \boldsymbol{u} \times \boldsymbol{v}$ as z-axis, see Figure 4.5(c). Next, four measures are calculated for each point pair using the angles between the points normal $\boldsymbol{n_s}$ and $\boldsymbol{n_t}$ and the distance vector between $\boldsymbol{p_s}$ and $\boldsymbol{p_t}$. The PFH feature is generated by subdividing the value range of the four measures and accumulating them in a multi-dimensional histogram. The



(a) LSP (b) THRIFT (c) PFH

(d) FPFH (e) SHOT (f) ECSAD

Figure 4.5: Geometric Attribute histogram

dimension of the PFH histogram is $\boldsymbol{div^4}$ where $\boldsymbol{div}$ is the subdivision factor. If each measure is split in to two bins, the histogram will have a size of $\mathbf{2^4 = 16}$. PFH is computational expensive and Rusu *et al.* [RBB09] excluded one dimension from the PFH descriptor (point distance between $\boldsymbol{p_s}$ and $\boldsymbol{p_t}$) in order to increase performance. Hence, the dimension is $\boldsymbol{div^3}$. In the same paper the author introduced another feature which is a faster and optimized version of PFH named "Fast Point Feature Histogram" (FPFH) [RBB09]. FPFH is a hierarchical feature which first computes a Simplified Point Feature Histogram (SPFH) for each feature point in the support region. In a second step, a weighted sum of the SPFH of the feature point and the SPFHs of the points in the support region to construct the final FPFH descriptor. Each SPFH are computed by calculating the three measures from the optimized PFH stated above between the feature point and its neighbours. The difference from PFH is that the measures are computed from the feature point of interest to all its neighbours and not between all points in the support region. The three measures are then accumulated in three separate histograms along the three feature dimensions and then concatenated into one FPFH histogram, see Figure 4.5(d). The dimension of FPFH is three times the number of bins along each dimension. In the standard implementation in Point Cloud Library **11** bins are used with a total FPFH feature dimension at 33 whereas the PFH implementation uses 5 bins which result in a dimension of $\mathbf{5^3 = 125}$.

The Signature of Histogram of OrienTations (SHOT) [TSDS10],[STS14] is together with Unique Shape Context (USC) of the same author, one of the first

local shape features to use unique and repeatable Local reference frames (LRF). The descriptor first constructs a LRF for the keypoint $\boldsymbol{p}$ and all its neighbours are aligned with this LRF. The support region is then divided into spatial 3D spherical volumes which are divided along the radial, azimuth and elevation axes, see Figure 4.5(e). A local histogram for each volume is constructed by computing the angles difference between the normals at the neighbouring points within the volume and the normal of the keypoint. This measure is accumulated into bins. The final SHOT descriptor is constructed by concatenating all local histograms. The size of the final SHOT descriptor is determined by the bin division along the radial, azimuth and elevation axes of the local histogram. In the PCL implementation the histogram has a size of 352 which is a relative large feature vector. The benefits of SHOT are that it is highly descriptive, computationally efficient and robust to noise. However, it is sensitivity to varying point densities. The same author proposed the Color-SHOT [TSS11] which combines the SHOT shape histogram with a texture histogram with a texture-related measure. Experimental results showed better performance in detection of highly similar shaped object but with different color.

One of the recent proposed descriptors is the Equivalent Circumference Surface Angle Descriptor (ECSAD) [JBK15]. This descriptor was originally developed for the purpose of detecting edges at orientation discontinuities in point clouds. First a LRF is defined from the eigenvectors of the scatter matrix. Next, the support region is divided in to volumes along the radial and azimuth axes, but not elevation which is the case for the SHOT feature, see Figure 4.5(f). Hence, the dimension of the feature and the probability of empty regions are decreased. For each of the neighbouring point in the volume the relative angle between keypoint normal and direction vector from the keypoint to the neighbouring point are computed. All angle measures are mapped in to spatial bins and each bin is averaged. In order to not have any empty bins an interpolation scheme is employed to fill in missing values in empty spatial bins. The ECSAD feature is efficient in feature matching due to the relatively low dimension of 30.

Deliberately, only significant work is included in this review of 3D local shape descriptors. A comprehensive review is given in [GBS+14].

4.4 Contributions

This section gives a short overview of the contributions to single camera pose estimation.

[Contribution E], entitles *Teach it Yourself - Fast Modeling of Industrial Objects for 6D Pose Estimation* is published on the 10th International Conference, ICVS 2015 held from July 6-9, 2015, in Copenhagen, Denmark.
The paper present a vision system for fast a easy modelling of Industrial objects and evaluate the performance of these object models in a 3D pose estimation pipeline.

[Contribution F], entitles *A large-scale 3d object recognition dataset* is submitted to the 4th International Conference on 3D Vision, which will be held at Standford University from October 25th-28th, 2016. The review is a double-blind review process. Paper notification date is the 31th of August 2016.
The paper present a new large scale dataset for 3D object recognition and a evaluation of state-of-the-art local shape features.

Teach it Yourself - Fast Modeling of Industrial Objects for 6D Pose Estimation

Thomas Sølund[1,3(✉)], Thiusius Rajeeth Savarimuthu[2], Anders Glent Buch[2], Anders Billesø Beck[1], Norbert Krüger[2], and Henrik Aanæs[3]

[1] Center for Robot Technology, Danish Technological Institute, Odense, Denmark
{thso,anbb}@dti.dk
[2] Mærsk Mc-Kinney Møller Institute, University of Southern Denmark, Odense, Denmark
{trs,anbu,norbert}@mmmi.sdu.dk
[3] Department of Applied Mathematics and Computer Science, Technical University of Denmark, Kongens Lyngby, Denmark
aanes@dtu.dk

Abstract. In this paper, we present a vision system that allows a human to create new 3D models of novel industrial parts by placing the part in two different positions in the scene. The two shot modeling framework generates models with a precision that allows the model to be used for 6D pose estimation without loss in pose accuracy. We quantitatively show that our modeling framework reconstructs noisy but adequate object models with a mean RMS error at 2.7 mm, a mean standard deviation at 0.025 mm and a completeness of 70.3 % over all 14 reconstructed models, compared to the ground truth CAD models. In addition, the models are applied in a pose estimation application, evaluated with 37 different scenes with 61 unique object poses. The pose estimation results show a mean translation error on 4.97 mm and a mean rotation error on 3.38 degrees.

Keywords: 3D modeling · Pose estimation · Robot manipulation · Flexible automation

1 Introduction

European manufacturing industries are challenged due to high wages, a growing number of product variants as well as a need for product customization. These facts imply an increasing demand for agile and flexible manufacturing systems. Especially, small batch sizes is changing the production paradigm from mass production to high/mix low/volume production [1]. This shift has changed the requirements to automation and industrial robotic systems, where e.g. high flexibility, reconfigurability and fast programming time are demanded.

The research leading to these results has been funded by the Danish Ministry of Science, Innovation and Higher Education under grant agreement #11-117524. and CARMEN under grant agreement #12-131860.

L. Nalpantidis et al. (Eds.): ICVS 2015, LNCS 9163, pp. 289–302, 2015.
DOI: 10.1007/978-3-319-20904-3_27

This paper presents a vision system that enables fast learning of geometrical object models in a multi view camera set-up intended for perception tasks, such as pose estimation and object recognition. With three pre-calibrated stereo camera pairs covering the scene a point cloud model of the object is extracted. The object is then turned manually to cover previously occluded parts of the surface followed by surface registration and post processing, to complete the model.

Geometrical modeling of industrial objects combined with 6D pose estimation based on visual information facilitates the reconfiguration of a robotic systems by reducing the effort for precise positioning. Vision guided robot systems in industry have until now been dominated by 2D or 2.5D vision solutions, which are hard to handle by users without expert vision knowledge. This has three reasons in particular: First, the viewpoint of the cameras need to be adapted to the stored views of the object. Second, often a rather awkward process is required to make sure that the space of required viewpoints is sufficiently densely sampled with object views often requiring large amounts of training data. Third, methods based on 2D pattern matching require a cumbersome extrinsic camera calibration process to be able to compute a 3D object pose.

In this context, 3D object models serve as a suitable abstractions of the general perception problem by lowering the training efforts needed compared to 2D or 2.5D vision applications and increasing the flexibility of the vision system. One common way is to use 3D CAD models [2]. Certain steps are needed in order to prepare a mechanical model, before it can be applied in a typical pose estimation pipeline. Typical, CAD models of the objects are (1) converted to a proper CAD file format (2) loaded into the vision system and (3) (down)sampled to get a point cloud representation with a correct point resolution by rendering and ray casting different views. Each step requires vision knowledge to parametrize correctly. The point resolution of the model has to roughly match with the point cloud resolution from the scene, that is directly dictated by the scene cameras. Furthermore, pose estimation algorithms need parametrization to fit the scene resolution. Creating the object representation directly by utilizing the scene cameras removes all before mentioned steps, thus reducing the (re)configuration time. Occasionally, there is no 3D CAD model of the object, if (1) the batch size is small, (2) the manufacturing company is small, (3) the particular object is customized, or (4) only 2D technical drawings of the object exist. Typically, it is time consuming to design the part in a CAD program, thus online 3D modeling is an appealing technology. Furthermore, existing CAD models are sometimes inaccurate, contain errors and/or lack important features which make them unsuitable for 6D pose estimation.

Hence fast and intuitive methods to train a robot system to recognize objects and estimate their poses are wanted [3,4]. In this paper, we present a two shot learning method for 3D models and apply it to pose estimation. Figure 1 shows our reconstruction pipeline that extracts the object from the scene with supervoxel segmentation and clustering followed by a registration pipeline that registers two partial models into one full model.

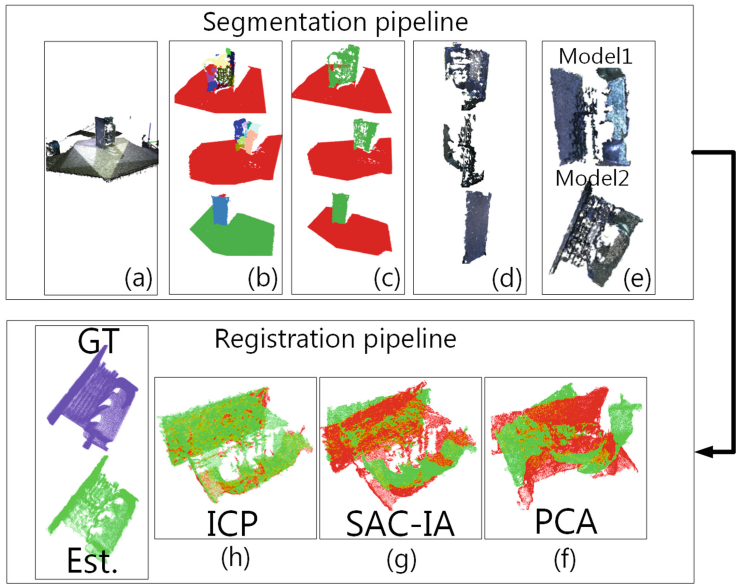


Fig. 1. Modeling steps: (a): *Step 1* - Transformation to camera frame and region-of-interest filtering of each view. **(b):** *Step 2* - Over segmentation of the scene. **(c):** *Step 3* - Clustering. **(d):** *Step 4* - Smoothing, up-sampling and filtering of each cluster. **(e):** - Output of the segmentation pipeline - two partial object models taken after the user has turned the object, **(f), (g):** *Step 6* -Initial alignment with Principal component analysis or SAmple Consensus - Initial Alignment. The initial registration method applied, is selected by the user. **(h):** *Step 7* - ICP registration. The estimated model (**Est.**) is shown together with the ground truth model (**GT**)

The main contribution of our work can be summarized as follows. **(a)** we present a novel multi-view two shot modeling framework suited for fast on-site modeling of industrial objects. **(b)** we present a comparison of our object models with the ground truth CAD model with respect to completeness and accuracy. **(c)** we evaluate and compare pose estimation results by applying our object models and ground truth models.

This paper is structured as follows: Sect. 2 contains related work; Sect. 3 describes our approach to realize two shot 3D model acquisition together with a description of our multi-view robot platform. Our evaluation protocol is outlined in Sect. 4, followed by a presentation of the results in Sect. 5. We conclude the paper in Sect. 6.

2 Related Work

Accurate 3D model reconstruction from a visual representation originates from reverse engineering science where object models are needed for computer animation, methodology, quality inspection etc. Initially rotatory tables are used for moving the object in front of a sensor to get range images form each view [5],

later robots are introduced to move the sensor or the object. A review of early work is presented in [6]. This review only considers geometric 3D model learning strategies of rigid objects with a robot. Learning object representation can basically be divided into four sub categories; (1) modeling by physical manipulation [7–9], (2) single view modeling using shape prior [3,10–12] (3) surface registration of multiple views [13–16] and (4) multi-view modeling with 360 degree scene coverage [17]. Our work belong the latter.

Modeling by Physical Manipulation: *Ilonen et al.* [7] fuse visual and tactile data to reconstruct a complete 3D model by grasping an object. Visual data from a single view RGB-D camera and a gripper padded with tactile sensors are fused using an iterative extended Kalman filter. The reconstruction method assumes that the objects are symmetrical. *Björkman et al.* [8] use Gaussian process (GP) regression to model an implicit surface of an object from a single Kinect view followed by tactile touch. The GP uncertainty is used to guide the robot to touch the model at areas with highest uncertainty. In [9] textured objects are modelled by detecting and tracking piecewise planar surfaces patches. Surfaces are merged and split into separate 3D object models during pushing actions with a robot manipulator. The method requires detecting interest points on the object surface.

Single View Modeling Using Shape Prior: In many applications, planar or rotational symmetry of an object can be assumed, but estimating symmetry axis is computationally hard because of the large search space and limited data available from a single view. *Bohg et al.* [10] bootstrap the search by limiting the set of hypothesis by only considering a vertical axis perpendicular to a plan. *Marton et al.* [3] fit geometric primitives (Boxes and cylinders) to the data taken from a single view. Additionally, a RANSAC based method for detecting the symmetry axis and completing surfaces of revolution is presented. For some objects detecting symmetry is error prone and a method such as point cloud extrusion [11] is a complementary method. Instead of detecting the symmetry axis and mirror the point cloud, fitting superquadrics [12] or implicit surfaces [8] have shown promising results.

Surface Registration of Multiple Views: Estimating the complete geometric representation of objects without making inference on shape geometry e.g. symmetry axis, requires that a sensor is either moved around the object [13,15] or the object is lifted by a robot and rotated in front of a camera [14,16] to cover unseen areas. *Bone et al.* [13] combine 2D silhouette based modeling and laser stripe scanning to estimate the shape of an object. In [15] a robot with a range sensor in the hand explores a table with objects. After first scan the object of interest is lifted, rotated and placed on the table and scanned using the same robot motion. *Kraft et al.* [14] present a sparse model representation based on 3D primitives composed of edge, line, orientation, phase and color transition for interest points. Their object learning framework grasps objects in the scene with a simple grasping reflex based on 3D primitives. When a grasp succeed the object is rotated in front of a stereo camera to accumulate 3D primitives. This results

in a sparse 3D representation of the model. *Krainin et al.* [16] show the same approach, but model the object as a dense surfel representation with a RGD-B camera. They introduce the articulated ICP which combines tracking of both the object and the robot manipulator while creating the model. The method combines a Kalman filter and ICP in an unified estimation process. With this approach they model symmetric objects, which in the case of general ICP is error prone due to ambiguities in object matching.

Our proposed method avoids some of the disadvantages of above mentioned methods e.g. no prior assumptions regarding object shape like symmetry [10] is required nor does our method fit geometrical primitives [3], superquadrics [12] or implicit surfaces [8]. Instead, the proposed method utilizes a multi-view calibration to align point clouds to capture the shape. With this, registration of different scans are avoided as in [14–16]. This enables us to model symmetric objects like cylinders and spheres with no texture which is difficult with classical registration of views, e.g., with the ICP algorithm. As the modeling framework takes two shots of the object, previous unseen surface patches, e.g., the bottom, are represented which is difficult with, e.g., single view modeling and fitting techniques without prior assumptions.

3 Two Shot 3D Object Modeling

The object modeling and all experiments are conducted on our experimental platform, consisting of two Universal robots[1] manipulators. The platform has three sensor clusters, each with a Microsoft Kinect v1 sensor, a BumbleeBee 2 stereo camera[2] and a XWGA projector for applying artificial texture to the scene in order to reconstruct industrial metallic objects. The three camera clusters give a 360-degree coverage of the scene and enables us to get a complete scene representations in one shot. Note, that only the BumbleBee stereo cameras and the XWGA projectors are used in this work. The platform is shown in Fig. 2 together with all 14 objects.

The extrinsic calibration of the platform is conducted in a single step multi camera and robot calibration procedure. As input for the calibration procedure, a rough model of the setup is needed including start guess on the robot and camera placements and the intrinsic calibration of the cameras. This allows the robot to automatically plan motions and move to valid positions where the cameras are able to detect a marker mounted in the tool of the robot. The extrinsic parameters of the camera and robot and the intrinsic parameters of the robot are computed using a set of detected marker poses and robot poses. In order to gain the required precision we calibrate two camera clusters at a time and let the robot move closer to these cameras. Based on a sample size of 402 corresponding images and robot poses we use 75 % of the samples to calibrate and the 25 % of the samples to verify the results. The residual of the calibration process shows an average error 2.8 mm in translation and 1.0 deg in rotation.

[1] Universal Robots - http://www.universal-robots.com/en/.
[2] http://www.ptgrey.com/bumblebee2-firewire-stereo-vision-camera-systems.

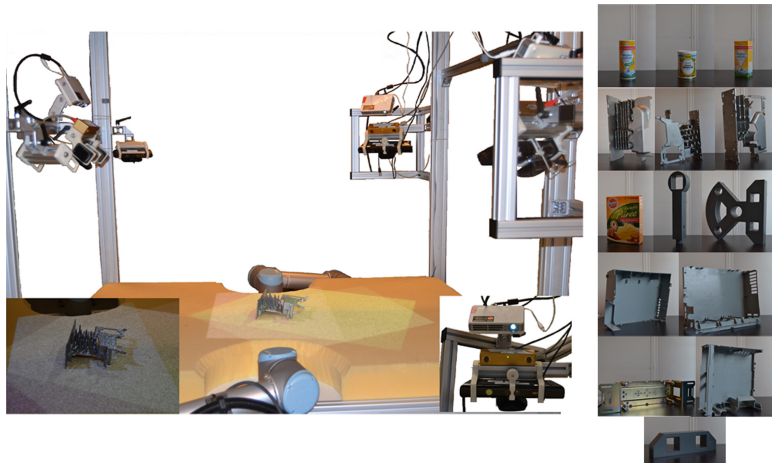


Fig. 2. Left image: The experimental platform. **Right image:** The 14 test objects.

Initially, the scene is reconstructed using Stereo Block Matching from OpenCV[3]. The reconstructed point clouds are aligned with the world frame located in the base of robot 1 by applying the extrinsic platform calibration, This allows the robot to grasp detected objects.

Extracting models from a 3D scene is basically a segmentation problem where points associated with the model have to be filtered. In robotics, traditionally it has been assumed that the object of interest is located on a dominant plane e.g. a table. The typical segmentation pipeline that has been used to remove the plane from the point cloud includes e.g. a RANSAC plane fitting algorithm followed by an Euclidean clustering algorithm. However, this pipeline is not optimal when it comes to model extraction because the plane removal algorithms, removes points in the transition between the object and the plane that belongs to the object. This fact is critical in our system, because a surface to surface registration is needed to finalize the two partial models into one full model. If the partial models lack correlated features, the registration of the surfaces is likely fail. Therefore we propose a different pipeline based on supervoxel segmentation proposed in [18] followed by a learning free segmentation algorithm that evaluates each supervoxel based on a local convexity criterion [19]. We extend this with a geometrical clustering algorithm based on a segment-to-plane and Euclidean distance criterion. In the following, we will outline the segmentation pipeline followed by the post-processing and surface to surface registration pipeline. An overview of the processing pipeline is given in Fig. 1.

3.1 Segmentation Pipeline

Our segmentation algorithm starts with segmenting each point cloud from the three views into supervoxels by applying the algorithm from [18] with a voxel

[3] OpenCV Stereo Block Matching: http://docs.opencv.org.

resolution equal 0.0035 and a seed resolution equal 0.020. The importance of color λ, spatial distance μ and normal direction ϵ in the computation of the seed expansion distance measure, is set to 2.0, 5.0, 8.0, respectively. A prerequisite for the algorithm in [19] is that the object is represented as a continuous point set and not containing discontinuities and large holes with missing points. The fact that we are reconstructing industrial objects with discontinues surfaces e.g. heatsinks with specular surface properties, will result in the algorithm not segmenting the object as one object. As a consequence we deliberately over-segment the scene into partial object segments and cluster these segments into one object representation. The algorithm segments the scene by considering the inclination angle of the super voxel normals direction to determine if the edge between two super voxels in the adjacency graph is concave or convex. The threshold that determines if an edge between two supervoxels is convex is set to 15.0 degrees and we remove supervoxels segments smaller that 10 to avoid noise.

The convexity based supervoxel segmentation algorithm described above results in an over-segmentation of the objects as shown in Fig. 1(b). We cluster the partial object segments into one object by examining the normal direction of the plane fitted each object segment. If the spherical inclination angle between the normal of the segment and the dominant plane in the scene e.g. a table, is larger than α and the plane-to-plane euclidean distance of the segment, is larger than β we accept the segment as a part of the object. We set α to 0.5 degrees and β to 5 mm. The clustering step is depicted in Fig. 1(c).

A post-processing step is conducted to refine the object model by smoothing and noise filtering. The surface is filtered by removing outliers based on two metrics; a statistical and a radius. The statistical outlier removal filter removes noisy measurement from the point cloud by considering the mean point distance in a local neighbourhood and remove points with a distance larger than a threshold. For removing spurious point clusters from the scene we filter the point cloud by looking in a radius around a point. If a point has less than 40 neighbouring points in a radius of 10 mm, the point is removed. The surface filtering is followed by an euclidean clustering step that select the largest cluster. A final step up-samples and smooth the surface by applying a moving least square filter with voxel grid dilation. The entire segmentation pipeline is implemented with use of the Point cloud Library[4].

3.2 Surface to Surface Registration

The segmentation pipeline extracts a partial object model that misses e.g. the bottom of the object. For covering previously occluded parts of the surface, the object is turned manually by the user and the segmentation pipeline is processed again which results in two partial object models, Fig. 1(e). The fact that the robot platform provides a full scene coverage gives us enough correlating object points between the two partial object models to register the two surfaces to one coherent object model. Initially, we roughly align the two object frames by computing the

[4] Point Cloud Library: http://pointclouds.org.

centroid and principal component of each model with PCA analysis, Fig. 1(f). The quality of this alignment is highly dependant on the object surface. The PCA analysis of pseudo-symmetrical objects with ambiguity tends to rotate the two object models differently. In order to cover this, the user has the possibility to initiate an additional alignment step before ICP registration. This step computes a new initial alignment based on shape features, in case the PCA misaligns the two partial object models. We adapt the SAC-IA method from [20] that compute an initial transformation for aligning the two surfaces by means of Fast Point Feature Histogram (FPFH) and SAmple Consensus, Fig. 1(g). The SAC-IA alignment is followed by a final ICP registration step, Fig. 1(h).

4 Evaluation Protocol

Two different experimental evaluations of the approach are presented. In Sect. 4.1 we outline the comparison of the object model obtained from our two shot modeling framework with the ground truth models, in terms of accuracy and completeness. This gives us a measure of how similar the object models are compared to the ground truth. In Sect. 4.2 we outline the protocol for testing and evaluating our models in pose estimation of industrial objects on our multi-view robot platform, presented in Sect. 3.

For evaluating the proposed method, the 14 different objects in Fig. 2 are reconstructed and categorized into three categories according to their surface properties. A sample of 4 of the 14 objects is illustrated in Fig. 3. The 14 objects include industrial parts with different surface properties as textured, non-textured, specular, non-specular, light and dark objects. Furthermore, some geometrical simple objects with few discriminative shape features, e.g., a cylinder and objects with many discriminative shape features are included. The objects are categories as following: (1) textured objects e.g. food containers with labels (Fig. 3 first row), (2) non-textured objects e.g. plastic parts for final assembly (fourth row) and (3) complex objects which posses some specular surface properties and high dynamic range (second row).

For evaluating the reconstructed model M_{est} with ground truth CAD model M_{gt} it is required that the two models are aligned to a common object frame. The alignment of the two models follows three steps, (1) manual selecting correspondences in M_{est} and M_{gt} (2) computing alignment transform that aligns M_{est} and M_{gt} using RANSAC and (3) run 500 ICP iterations with decreasing correspondence distance, to compute the final (rigid) alignment transform T.

4.1 Model Comparison

Comparing 3D models has formerly been conducted as a mesh to mesh comparison of watertight surfaces. Metrics like Hausdorff distance or Mean Square Error (MSE) are applied in order to compute an error map. This surface-to-surface methodology can be error prone when an accurate point-to-point error is required due to the natural vertex modification step in many reconstruction

algorithms e.g. Possion or Marching Cube algorithms. Instead a point-to-point measure is applied to avoid introducing a reconstruction error term arising from vertex modification and choice of reconstruction parameter. We compare the point cloud model with the ground truth CAD model by computing the number of correct points and the point accuracy of the estimated object models. For each 14 models the accuracy and completeness is computed as an evaluation measure for the model quality, where:

- **Accuracy:** is measured as the root mean square distance from each Point $\boldsymbol{P_{est}}$ in the estimated model $\boldsymbol{M_{est}}$ to the nearest neighbouring point $\boldsymbol{P_{gt}}$ in the ground truth model $\boldsymbol{M_{gt}}$. This measure expresses the quality of the reconstructed point $\boldsymbol{P_{est}}$ in $\boldsymbol{M_{est}}$.
- **Completeness:** is measured as a percentage of correct reconstructed points in $\boldsymbol{M_{est}}$. We compute the root mean square distance from each Point $\boldsymbol{P_{gt}}$ in the ground truth model $\boldsymbol{M_{gt}}$ to the nearest neighbouring point $\boldsymbol{P_{est}}$ in the estimated model $\boldsymbol{M_{est}}$. We threshold this distance and count the number of correct points. The threshold is empirical chosen to be 3 times the average point resolution of the scene.

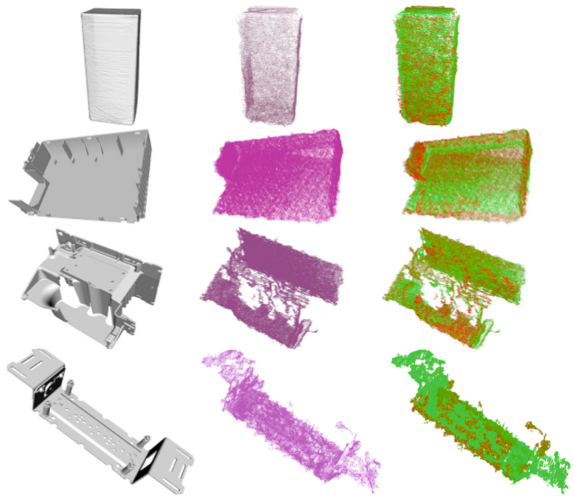


Fig. 3. Samples of the objects used in the evaluation. **First column:** Ground truth CAD model of the objects. **Second column:** Estimated models. **Third column:** Error model showing the noise distribution compared to ground truth.

The accuracy is reported in Sect. 5 as the average median error $\tilde{x}$, the standard deviation σ and RMS error $\boldsymbol{M_{rms}}$ of probability distribution functions (PDFs). We report the median error $\tilde{x}$ of the PDF to have a measure that is not biased by large outliers.

4.2 Pose Estimation

In order to quantify that our method is a valid visual learning method for computing 3D object representations, we evaluate the performance of a pose estimation algorithm with our estimated models and the ground truth models. This is to verify that learned object models result in an adequate pose accuracy and recognition rate compared to ground truth.

The objective of pose estimation is to estimate the (rigid) Transformation $\boldsymbol{T} \in \boldsymbol{SE(3)}$ that minimizes the mean square distances between each point $\boldsymbol{P_{model}}$ in an object model and the corresponding point $\boldsymbol{P_{scene}}$ in the scene. For the evaluation we have recorded 37 different multi-view scenes. For the evaluation we use 6 objects, paired three by three in two different sets of scenes. Each set consists of scenes with one, two and three objects such that all objects are represented by themselves and together with one or two other objects. Each set consists both of scenes where objects are touching each other and scenes where they are not. All scenes are recorded with the world frame in robot 1 base as global coordinate system. Ground truth data is obtained for all 37 scenes by manual annotation of 4 point in each point cloud, computing initial transformation followed by a large number of ICP iterations as described in Sect. 4.

We use a classic RANSAC 'hypothesis and test' algorithm together with Fast Point Feature Histogram (FPFH) local features [20]. For each scene and object, normals are estimated followed by feature estimation and RANSAC pose estimation. For all 6 models we measure the recognition rate for both the generated model and the ground truth. Each model is considered correct recognized if the translation t_r and the rotation R_r of the resulting pose P_r follows

$$t_r = ||t_{gt} - t_p|| \quad < 10\,mm \tag{1}$$

$$R_r = \arccos \frac{trace(R_p^T R_{gt} - 1)}{2} \quad < 10^\circ \tag{2}$$

where t_{gt} and R_{gt} are the ground truth translation vector and rotation matrix and t_p and R_p are the estimated translation vector and rotation matrix. We use 10 mm as threshold value because all objects in all 37 scenes have a distance of more than one meter from the sensor. With a Bumblebee stereo camera with a focal length equals 1320 pixels and baseline equal 0.12 m, a one pixel disparity error will result in a 6 mm depth error at one meter distance, thus a 10 mm pose error threshold is a good compromise. In addition to the recognition rate the pose accuracy of the models are determined by computing the translation error from Eq. 1 and the rotation error from Eq. 2. The results are presented in Sect. 5.

5 Results

In Table 1 the results of the model comparison are presented. Our method reconstructs the 14 models with a completeness ranging from approximately 52 % to 87 %. The low level of completeness for some objects e.g. the *Pendulum* and

Angular_bracket has different reasons. In case of the *Pendulum*, the object suffers from an incomplete registration due to object ambiguities and the fact that the surface of the *Angular_bracket* has a highly reflecting surface which results in missing points in the reconstruction. In these cases, the system reconstructs objects with a lower level of completeness. On the other hand, objects with good surface properties and less ambiguities are reconstructed with 70–85 % correct surface points by our system, with a satisfying accuracy when compared to the scene point resolution and noise level.

Table 1. Results from comparing model M_{est} with the ground truth M_{gt}

Results models vs. ground truth				
Model ID	RMS error	Stddev σ	Median $\hat{x}$	Completeness %
Marmalade*	3.43e-3	7.02e-6	2.31e-6	79.57
Salt Box*	2.04e-3	3.81e-6	0.84e-6	85.87
Salt Cylinder*	2.55e-3	9.77e-6	2.79e-6	86.82
Potato box*	1.57e-3	5.89e-6	5.28e-6	78.59
Rear Part A1$^{\#}$	2.73e-3	57.1e-6	0.48e-6	75.69
Rear Part A2$^{\#}$	3.98e-3	86.2e-6	0.83e-6	82.16
Rear Part A3$^{\#}$	1.05e-3	2.57e-6	0.54e-6	80.63
Heatsink A1$^{\#}$	1.52e-3	6.35e-6	0.62e-6	68.41
Heatsink A2$^{\#}$	4.82e-3	9.64e-6	0.46e-6	58.19
Heatsink A3$^{\#}$	1.47e-3	8.38e-6	0.88e-6	60.77
Faceplate^{+}	4.89e-3	74.7e-6	2.27e-6	54.54
Pendulum^{+}	2.68e-3	11.6e-6	1.71e-6	55.91
Seperator^{+}	2.96e-3	29.2e-6	2.23e-6	52.19
Bracket$^{\#}$	2.63e-3	35.7e-6	1.19e-6	64.46
Average	2.67e-3	24.8e-6	1.60e-6	70.27

objects from real industrial production sites, + Cranfield benchmark objects
*from KIT object database http://i61p109.ira.uka.de/ObjectModelsWebUI/

The evaluation of the useability of the reconstructed models for pose estimation is conducted with a RANSAC pose estimation algorithm. The RANSAC algorithm runs for 5000 iterations with an inlier fraction at 0.2. The scene and the model are down-sampled to 5 mm and the pose estimation algorithm runs with an Euclidean inlier threshold at two times the scene point resolution. Each pose estimate is followed by a pose refinement step with 200 ICP iterations.

Our dataset consists of 37 different scenes with 61 unique object poses. We correctly estimate the pose of 38 object with our model and 33 with the

ground truth model which result in a recognition rate at 62 % and 54 %, respectively. The quit low recognition rate is related to the difficulty of the scenes with many similar objects placed closely together or on top of each other. In Fig. 4 the pose error obtained from the reconstructed and the ground truth CAD model are presented as histograms. The results show that our models and ground truth are performing equal but our models have a slight lower average rotation error at 3.38 degrees, measured on all estimated poses of recognized objects. The ground truth has a slightly larger rotation error on 3.94 degrees. On the other hand, the ground truth has a lower average translation error on 3.0 mm compared to the estimated models having an average error on 4.97 mm. The overall conclusion is that the method has reconstructed the models with a completeness between 52 % to 87 % and a mean point accuracy between 2.04 to 4.89 mm. The result of the pose estimation evaluation shows that using the estimated models one get approximately the same pose accuracy than using ground truth CAD models. This result is satisfying for robot manipulation. The estimated models have a

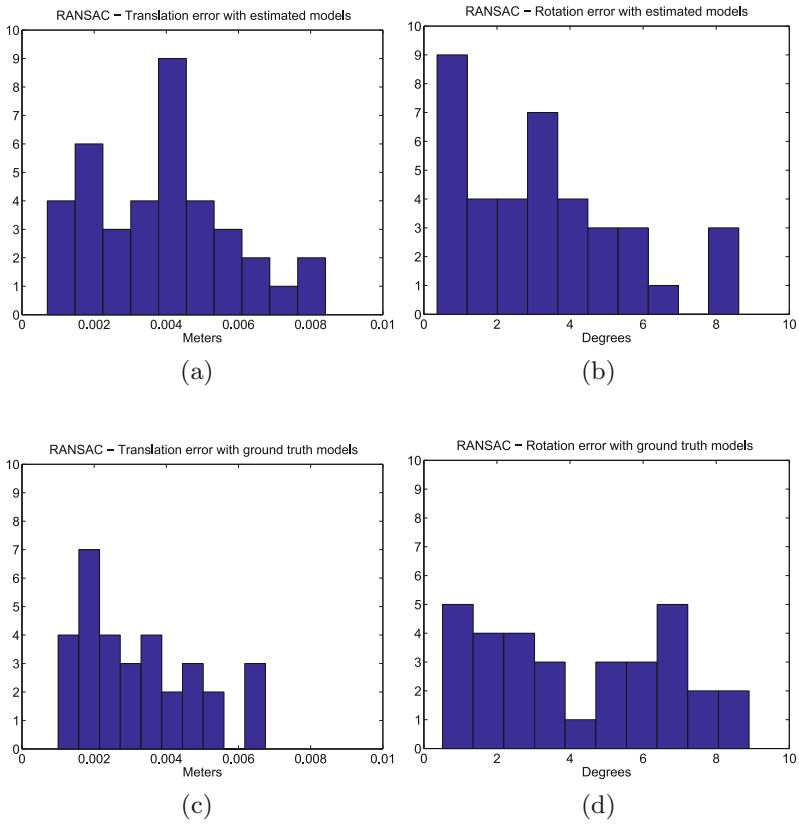


Fig. 4. Histogram of the pose error of all accepted pose estimates based on the criterion in Eqs. 1 and 2. **(a)** translation error of the pose estimation results with the reconstructed model. **(b)** rotation error with the reconstructed model. **(c)** translation error with the ground truth model. **(d)** rotation error with the ground truth model.

slightly better recognition rate than compared to ground truth. This could imply that having models which posses some of the scene characteristics in terms of noise levels and distortions of object borders, actually could improve the recognition rate abit. A more realistic representation of the object results in better surface normals and computed features that in the end favours pose estimation algorithms. However, to determine this correlation a larger dataset is required. This will be investigated in future work, together with a study concerning dexterous grasp simulation in combination with the reconstructed models.

6 Conclusion

We presented a multi view vision system able to reconstruct full 3D object models in only two shots. Our experiments show that object models larger than $(7.5 \times 7.5 \times 7.5)$ cm are reconstructed with an adequate accuracy and completeness. Furthermore, the models are useful in 6D pose estimation applications, without loss in recognition rate and precision compared to the ground truth CAD model. The combination of a 360-degree scene coverage from three calibrated stereo pairs and the two shot modeling methodology make the method useful for flexible reconfiguration of vision systems in industry. This flexibility makes the approach suited for few-of-a-kind production in industry where many new novel objects have to be handled by a robot thus reconfigurable vision systems reducing the set-up times.

References

1. Bannat, A., Bautze, T., Beetz, M., Blume, J., Diepold, K., Ertelt, C., Geiger, F., Gmeiner, T., Gyger, T., Knoll, A., Lau, C., Lenz, C., Ostgathe, M., Reinhart, G., Roesel, W., Ruehr, T., Schuboe, A., Shea, K., Wersborg, I., Stork genannt Wersborg, S., Tekouo, W., Wallhoff, F., Wiesbeck, M., Zaeh, M.F.: Artificial cognition in production systems. IEEE Trans. Autom. Sci. Eng. **8**(1), 148–174 (2011)
2. Buch, A., Kraft, D., Kamarainen, J.-K., Petersen, H., Kruger, N.: Pose estimation using local structure-specific shape and appearance context. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 2080–2087, May 2013
3. Marton, Z., Pangercic, D., Blodow, N., Kleinehellefort, J., Beetz, M.: General 3D modelling of novel objects from a single view. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, pp. 3700–3705, October 2010
4. Mustafa, W., Pugeault, N., Kruger, N.: Multi-view object recognition using view-point invariant shape relations and appearance information. In: IEEE/RSJ International Conference on Robotics and Automation, Karlsruhe, Germany, pp. 4230–4237, May 2013
5. Chen, Y., Medioni, G.: Object modeling by registration of multiple range images. In: Proceedings of the 1991 IEEE International Conference on Robotics and Automation, Sacramento, California, pp. 2724–2729, April 1991
6. Bernardini, F., Rushmeier, H.: The 3D model acquisition pipeline. Comput. Graph. Forum **21**(2), 149–172 (2002)

7. Ilonen, J., Bohg, J., Kyrki, V.: Three-dimensional object reconstruction of symmetric objects by fusing visual and tactile sensing. Int. J. Robot. Res. **33**(2), 321–341 (2013)
8. Björkman, M., Bekiroglu, Y., Hogman, V., Kragic, D.: Enhancing visual perception of shape through tactile glances. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, November 2013
9. Prankl, J., Zillich, M., Vincze, M.: Interactive object modelling based on piecewise planar surface patches. Comput. Vis. Image Underst. **117**(6), 718–731 (2013)
10. Bohg, J., Johnson-Roberson, M., Leon, B., Felip, J., Gratal, X., Bergstrom, N., Kragic, D., Morales, A.: Mind the gap - robotic grasping under incomplete observation. In: IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, May 2011
11. Kroemer, O., Ben Amor, H., Ewerton, M., Peters, J.: Point cloud completion using extrusions. In: IEEE-RAS International Conference on Humanoid Robots (Humanoids), Osaka, Japan, pp. 680–685, November 2012
12. Duncan, K., Sarkar, S., Alqasemi, R., Dubey, R.: Multi-scale superquadric fitting for efficient shape and pose recovery of unknown objects. In: IEEE/RSJ International Conference on Robotics and Automation, Karlsruhe, Germany, pp. 4238–4243, May 2013
13. Bone, G.M., Lambert, A., Edwards, M.: Automated modeling and robotic grasping of unknown three-dimensional objects. In: 2008 IEEE International Conference on Robotics and Automation, Pasadena, California, pp. 292–298, May 2008
14. Kraft, D., Detry, R., Pugeault, N.: Development of object and grasping knowledge by robot exploration. IEEE Trans. Auton. Mental Dev. **2**(4), 368–382 (2010)
15. Aleotti, J., Rizzini, D.L., Caselli, S.: Perception and grasping of object parts from active robot exploration. J. Intell. Robot. Syst. **76**, 401–425 (2014)
16. Krainin, M., Henry, P., Ren, X., Fox, D.: Manipulator and object tracking for in-hand 3D object modeling. Int. J. Robot. Res. **30**, 1311–1327 (2011)
17. Olesen, S.M., Lyder, S., Kraft, D., Krüger, N., Jessen, J.B.: Real-time extraction of surface patches with associated uncertainties by means of kinect cameras. J. Real-Time Image Process. **10**(1), 105–118 (2012). doi:10.1007/s11554-012-0261-x
18. Papon, J., Abramov, A., Schoeler, M., Wörgötter, F.: Voxel cloud connectivity segmentation - supervoxels for point clouds. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, Oregon, 22–27 June 2013
19. Stein, S., Schoeler, M., Papon, J., Worgotter, F.: Object partitioning using local convexity. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 304–311, June 2014
20. Rusu, R., Blodow, N., Beetz, M.: Fast point feature histograms (FPFH) for 3D registration. In: IEEE International Conference on Robotics and Automation, ICRA 2009, pp. 3212–3217, May 2009

A Large-Scale 3D Object Recognition dataset

Thomas Sølund
Danish Technological Institute
DK-5230 Odense M, Denmark
thso@dti.dk

Anders Glent Buch, Norbert Krüger
University of Southern Denmark,
DK-5230 Odense, Denmark
anbu,norbert@mmmi.sdu.dk

Henrik Aanæs
Technical University of Denmark
DK-2800 Kgs. Lyngby, Denmark
aanes@dtu.dk

Abstract

This paper presents a new large scale dataset targeting evaluation of local shape descriptors and 3d object recognition algorithms. The dataset consists of point clouds and triangulated meshes from 292 physical scenes taken from 11 different views; a total of approximately 3204 views. Each of the physical scenes contain 10 occluded objects resulting in a dataset with 32040 unique object poses and 45 different object models. The 45 object models are full 360 degree models which are scanned with a high precision structured light scanner and a turntable. All the included objects belong to different geometric groups; concave, convex, cylindrical and flat 3D object models. The object models have varying amount of local geometric features to challenge existing local shape feature descriptors in terms of descriptiveness and robustness. The dataset is validated in a benchmark which evaluates the matching performance of 7 different state-of-the-art local shape descriptors. Further, we validate the dataset in a 3D object recognition pipeline. Our benchmark shows as expected that local shape feature descriptors without any global point relation across the surface have a poor matching performance with flat and cylindrical objects. It is our objective that this dataset contributes to the future development of next generation of 3D object recognition algorithms. The dataset will be made public available together with this paper.

1. Introduction

Object recognition from range images is a fundamental research area in computer vision with many different applications in different industries. With the continually introduction of new inexpensive 3D sensors for different applications, the ability to localize and recognize rigid and no rigid objects is an attractive and unavoidable technology.

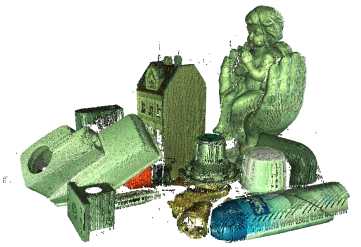


Figure 1: Example of a full registered scene included in the dataset

Applications areas such as robotic assembly, bin-picking, mobile robotic manipulation, biometric analysis, tracking and intelligent surveillance all benefits from 3D data for localizing objects. Mainly, because the third dimension is explicit given and not inferred as in 2D object pose estimation. In the last decades many contributions on 2D object recognition and classification have been published, where methods are evaluated on large-scale dataset like the PASCAL Visual Object Challenge (VOC) [10] and ImageNet [7]. The benefit from evaluating algorithms on a large-scale dataset has proven valuable in the continues improvement of recognition algorithms after the release of the PASCAL VOC and ImageNet datasets [10]. In 3D object recognition research, large-scale datasets that consist of a set of 3D querie models Q_n, 3D target scenes S_t and ground truth poses T_{gt} for each object in each scene are required in order to be able to evaluate existing and future 3D object recognition algorithms better. Until now, 3D object recognition algorithms are evaluated with eight smaller datasets [17], including in the UWA [27],[28], Queens [38],[38]

	M_q	S_t	Sensor	M_q in S_t	M_q normals	M_q mesh	S_t normals	S_t mesh	Full 6D pose	Occlusion	Clutter	[R]/[S]
UWA [27], [28]	5	50	LIDAR	(5)/(5)	✓	✓	✓	✓	✓	✓	%	R
Queens Lidar [39], [38]	5	80	LIDAR	(1-5)/(1-5)	✓	✓	%	✓	✓	%	%	R
Queens Stereo [39], [38]	5	100	Stereo	(3)/(3)	✓	✓	%	✓	✓	%	%	R
Bologna 1&2 [35], [42]	6	45	-	(3-5)/(3-5)	✓	✓	✓	✓	✓	%	%	S
Bologna 3 [35], [42]	8	15	Spacetime	(2)/(5-6)	✓	%	✓	✓	✓	%	%	R
Bologna 4 [35], [42]	8	16	Spacetime	(2)/(6)	✓	%	✓	✓	✓	%	%	R
Bologna 5 [35], [42]	6	16	Kinect V1	(2-4)/(5-9)	✓	%	✓	✓	✓	%	%	R
Vienna Kinect [2]	35	50	Kinect V1	(1-5)/(1-5)	✓	✓	%	%	✓	%	%	R
RGB-D Scenes V1 [24], [25]	5	8 videos	Kinect V1	(0)/(5)	%	%	%	%	%	%	%	R
RGB-D Scenes V2 [23]	9	14	Kinect V1	(0)/(9)	%	%	%	%	%	%	%	R
TÜM [30]	20	150	-	(3-5)/(3-5)	✓	✓	✓	✓	✓	✓	✓	S
Willow [43]	35	177	Kinect V1	*	✓	✓	%	%	%	%	%	R
ECCV12 [2]	35	50	Kinect V1	(3-7)/(3-7)	%	✓	%	%	✓	%	%	R
Alicante [14]	28	9	Kinect V2	*	%	✓	%	0	✓	%	%	R
Our dataset	**45**	**3204**	SL	(10)/(10)	✓	✓	✓	✓	✓	✓	✓	R

Table 1: Comparison of existing datasets for 3D object recognition with the presented datasets. M_q is the amount of different models in the dataset and S_t is the amount of scenes. M_q in S_t shows how many models annotated in each scene and how many objects there are in each scene (No. annotated in scene/No. of object in scene).[R]/[S] indicates whether the dataset is synthetic or acquired in the real world. *(* dataset unavailable online)*

and Bologna [35],[42] datasets. Other, smaller dataset like the Vienna Kinect[2], TUM [30], TUM-LineMod[19], BigBird[36] and RGB-D dataset version 1 & 2 [24],[23] are proposed. Common for all datasets is that the amount of scenes and/or models are limited.

In this paper we present a new large-scale dataset consisting of 45 objects and 3204 views. The new dataset is recorded systematically in an environment without ambient light and with controlled illumination. The system includes an industrial robot in a dark chamber, a high precision structured light scanner which records data of 292 different scenes from 11 different view points. Each scene consists of 10 occluded objects, automatically taken with a structured light sensor, which results in a dataset with 32040 unique object poses. With 11 different views of the same scene our dataset is especially suited for studying the effect of view point changes in 3D object recognition. The objects are configured as a classic table-top scenario where different objects are depicted form the side. The dataset is not meant as an evaluation platform for bin-picking and top-view scenarios where many instances of the same object are present in the scene. The table-top scenario is selected because it encapsulates many of the problems and challenges in 3D object recognition in favour of a larger research community. Our 45 object models are scanned in a similar dark chamber with a high precision structured light scanner and a rotation table. With this setup we are able to scan objects with an average point resolution of 275 microns. We validate our dataset by evaluating state of the art local shape features in a 3D object recognition pipeline.

Why does the 3D Object recognition community need another dataset? Previous proposed datasets are limited in the total number of objects, object per. scene and total number of scenes. Additional, the included objects in the different datasets are mostly geometrical ideal objects. With geometrical ideal objects we refer to objects with concave water tight and closed surfaces that are rich of local descriptive geometrical features, like the UWA-chef model [27],[28], the bologna-Armadillo [35],[42] and the Queens-BigBird [38],[38]. In our dataset we include not only ideal concave objects, but flat, cylindrical, feature-rich, simple concave and convex objects. It is expected that some local shape features will have a poor performance on our dataset because the lack of local geometric features. However, our main goal for proposing this dataset is to highlight the challenges in 3D object recognition research and strengthen the data foundation in future algorithm development. This work is initiated by our previous experiences on pose estimation of geometric simple objects in an industrial robotic context. We hope that new novel methods for 3D object recognition will emerge in the future, as a side-effect of this dataset. Not only local shape descriptors but to a great extend features that use point relations across the surface e.g. Point Pair Features [3], semi-global features and template matching. We have selected the different objects based on previous experience, as a combination of lab-objects and industrial objects. The industrial objects are provided by companies that need to pick the objects from boxes or bins with a robot. Thus, the dataset is a mix of real industrial objects that we know from experience is difficult to detect and objects from

our lab which are more suited for a 3D object recognition pipeline with local shape features.

This paper is structured as follows: In Section 2 related datasets and local feature descriptors are presented. Section 3 outlines our experimental design followed by Section 4 which presents our benchmarking methodology. In Section 5 the results are given followed by a discussion and conclusion in Section 6.

2. Related work

We will now relate our work to state of the art. First, we review existing 3D object recognition datasets followed by a concise review of significant local feature descriptors.

3D Object Recognition datasets:
The last 10 years a few 3D object recognition datasets have been published. Mainly in conjunction with algorithms for local feature description [27],[38],[39],[42],[35], correspondence matching and rejection [30], pose hypothesis verification [2] and 3D keypoint detection [28]. Common for all existing dataset are the limited number of 3D object models and scenes. A comparison of the different datasets are presented in Table 1, from which it is seen that our dataset is magnitudes larger. The most common used datasets are the UWA [27], [28], Queens [38],[39] and the Bologna [35], [42]. These datasets are widely used in performance evaluations of local feature descriptors [17], [4], keypoint detectors [34] and surveys [16], [11]. The main problems of all the datasets are; size, variety of objects, few objects per. scene and missing occlusion/clutter estimates. In this work we are not considering 2.5D recognition methods where either a full object model or sampled templates is used for recognizing object in a RGB-D image. However, some datasets exist for this problem e.g. the Line-Mod dataset [19] or the recent published RGB-D dataset for warehouse pick-and-place tasks [29].

Local feature descriptors:
During the last three decades, a vast number of different 3D local feature descriptors have been proposed including SPLASH [37], Spin Image (SI) [20], 3D Shape Context (3DSC) [13], LSP [5], 3D Tensors [27], THRIFT [12], MESH-HOG [44], ISS [45], Unique shape context (USC) [41], Point Feature Histogram (PFH) [32], Fast Point Feature Histograms (FPFH) [31], SHOT [42], ROPS [18], ECSAD [21] and Tri-Spin-Image (TriSI)[17]. The descriptors find usages in applications such as 3D object categorization, recognition, retrieval, analysis, registration and reconstruction among others. Designing descriptors which are distinctive and robust toward occlusion and noise is still an ongoing research topic.

Local feature descriptors aim at computing a distinctive and robust N-dimensional feature vector around a point, by considering the points in an Euclidean neighbourhood. The support radius determining the size of the neighbourhood is often one of the critical parameters in the pipeline. Local feature descriptors are often split into two different categories, spatial and geometrical histograms [35]. Recent studies have shown that the state of the art features are not generalizing well over many types of geometry classes, [17],[4]. The studies proved one of the main issues in 3D object recognition today, that there exist no local shape feature which describes the geometry well over many different object classes, e.g. flat, rotational symmetrical and geometrically feature rich objects. What feature(s) to use are still dependent on the object geometry. However, the experimental results in [4] showed the advantage of fusing several local feature descriptors. The studies from [17],[4] leave a relevant research question to be answered; are the local feature descriptors not descriptive enough to generalize different object classes or are the data used for evaluation too limited? The evaluation datasets used for the evaluation in general and in [17],[4] consist mainly of objects with many descriptive and distinctive geometrical features.

3. Experimental design

We have constructed a dataset by achieving highly accurate 3D model of each of the 45 individual objects, as described in Section 3.1. These are then used to compose 292 individual scenes where 10 objects are placed in different configurations. A robot moves our scanner to 11 fixed positions in order to create 11 independent view points of the same object configuration, as described in Section 3.2. Hence, we have 3204 individual observations of 10 objects included in the dataset. We argue that these observations are independent because of the large view point change at 36 degree horizontal and 45 degree vertical. Thus, the objects surface are very different in each view. We achieve very accurate ground truth poses by annotating the entire dataset with our high resolution object models in full resolution, as described in Section 3.3.

3.1. Object model scanning

Our object models are scanned with a high precision structured light setup consisting of two industrial cameras (Point Grey Research GS3-U3-91S6C-C) and a high resolution DLP projector (LG PF80G) mounted on a rigid aluminium frame [9]. In addition, a high precision turntable (Newmark Systems RT-5) is used in order to provide automatic rotation of the object. Each of the 45 objects are incremental scanned with a rotation of 20 degrees. All individual scan views are reconstructed by the Line shifting algorithm [15], which results in accurate and dense point clouds of the objects. Eleven temporal binary gray code

patterns are projected followed by eight line shifting patterns. The point resolution of the scanned objects are in average 275 microns and consist of around 908000 vertices and 1.8 million faces in average. Once a single view of the model is scanned, noisy outliers of the measurement are manual removed and surface normals are estimated to ensure consistent normals. All 18 views are registered with Iterative Closest Point (ICP) and a new object frame is computed with principal component analysis on the point set. All models are sampled with a Poisson disk sampling algorithm [6] and triangulated with the poisson reconstruction algorithm [22] with an octree depth=14, solver divide = 8 and iso divide = 5. We use the PCL implementation [33]. All models are provided as coloured point clouds and triangular meshes, all in the *.ply* format, see Figure 2. Note that the modelling setup is not radiometrical calibrated.

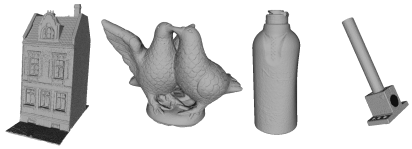


Figure 2: A sample of the scanned object models

3.2. Scene scanning

For the data collection we use a 6-axis ABB IRB 1600 industrial robot to provide a precise and highly repeatable camera pose. The robot is equipped with two PointGray Grasshopper3 GS3-U391S6C-C USB3 color cameras with resolution of 9.1 Mp and a Wintech Pro4500 projector with a resolution of 1140x912 pixels. The sensor cluster with the structured light sensor (SL) is mounted in the robot tool. The Robot setup is constructed as a radiometric "dead" which gives a scene representation with zero ambient light [1]. All scene illumination is controlled in the recording process. The sensor cluster is calibrated with an automatic calibration procedure which includes a stereo calibration of the SL sensor using `OpenCV`[1].An automatic hand eye calibration [8] is conducted in order to align the structured light scans in the world frame. The world frame is placed in the robot base frame. Each physical scene is scanned from 11 different views with the structured line scanner (SL). During the structured light scanning process the chamber is completely dark in order to increase the signal-to-noise ration of the scan. The structured light scans are reconstructed with the Line shifting algorithm [15] with ten temporal pattern levels. The views are distributed equally on a quarter sphere around the scene, such that each scene is depicted from - 90 to 90 degrees horizontally and 0 to 45 degree vertically. The distance between the sensor views and the center of the scene is 0.8 meter. In order to be able to reproduce the results we store all images in raw Portable Network Graphics in full resolution (9.1 MP). Debayering, white balance, image rectification, pattern decoding and point cloud reconstruction are a off line process.

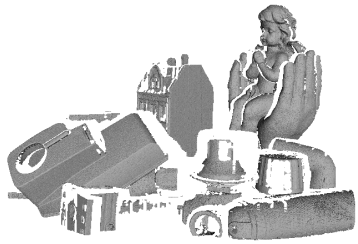


Figure 3: Example on one of the 3204 triangulated scene

Local shape features like 3D Tensor [27], ROPS [18] and FPFH [31] apply the underlying mesh surface during feature computation. In order for the dataset to support these algorithms all scenes are triangulated to a polygonal mesh. For each scene a set of corresponding triangles are computed with a 2D delaynay triangulation algorithm in VTK[2]. Each (x, y) point coordinates of the scene point cloud are normalized with its z component in order to project the 3D point cloud into 2D and triangulated. The 3D mesh structures are created by re-assigning all 3D point with the triangles of the same index. Long edges in the mesh are then removed form the mesh in order to avoid triangles between step edges. An example of a triangulated scene is shown in Figure 3.

3.3. Ground Truth 6D pose

For each of the 3204 scenes the 6D ground truth pose, occlusion and clutter estimates for each object are provided. We estimate the occlusion from Equation 1 and clutter from Equation 2 by counting the number of points at the object model in which the squared euclidean distance to a scene point is less than two times the scene resolution. Both the object model and the scene are sampled to ensure equal point distance.

$$Occlusion = 1 - \frac{visible\ object\ points}{total\ object\ points} \quad (1)$$

$$Clutter = 1 - \frac{visible\ object\ points}{total\ scene\ points} \quad (2)$$

Accurate ground true poses are ensured by manual annotation of each object in the scenes. First, one point cloud that covers 180 degree of the scene is stitched together form each of the 11 views and registered with ICP. This full scene

[1] `http://opencv.org/`

[2] `http://www.vtk.org/`

point cloud is applied in the annotation process, see Figure 1. The full scene point cloud covers the geometric structures of each model in the scene with more points compared to the individual views. Thus, it is possible to get a more accurate ICP registration of the object model in the scenes. All 290 combined scenes are manually annotated by selecting four identical points for each model and the scene, followed by an estimation of the rigid transform. An iterative ICP, which incremental decreases the allowed correspondence distance ensures a very accurate final ground truth pose of each object in the scene. The final ICP iterations are accomplished with the full resolution model to get the best fit of the model points to the scene points. The individual ground truth poses in each sensor view T_{sensor_n} is computed by transforming the ground truth poses from the world frame T_{world} to each of the individual sensor frames T_{sensor_n}. Equation 3 shows the final transformation.

$$T^{world}_{sensor_n} = T^{-1}_{robot} \cdot T^{-1}_{HandEye} \cdot T^{-1}_{icp} \cdot T_{gt} \qquad (3)$$

where T_{robot} is the known robot pose for each view point, $T_{HandEye}$ is the calibrated hand eye transform, T_{icp} is the alignment transform which align view n to view 0 in the world frame and T_{gt} is the annotated ground truth pose in the full scene point cloud with the world frame as reference. This methodology guarantees accurate pose in $Tsensor_n$ independent of the amount of occlusion. Even in views with limited number of scene points of an object the ground truth pose is accurate because the ground truth pose is computed in the full scene point cloud. For each ground truth pose in each sensor view, we compute the RMS error between the model and the scene to guarantee the overall accuracy of all ground truth poses. On average the RMS error of all ground truth poses is within 0.15 mm.

4. Benchmark

This section outlines the experimental protocol defined to validate the dataset. The protocol is inspired by Salti *et al*. [35]. The evaluation is divide into two parts; feature matching accuracy and object recognition rate. The selected local feature descriptors for our evaluation include; Spin Image (SI) [20], PFH [32], FPFH [31], USC [41], SHOT [42], ROPS [18], ECSAD [21] and NDHIST [4]. The features are selected based on implementation availability and results from previous studies on feature descriptor benchmarking [17],[4].

4.1. Feature Matching

The descriptiveness and accuracy of a feature descriptor are measured with Precision-Recall and presented as 1-Precision vs. Recall Cruves (PRC). First we sample both the query models and the target scenes with a voxel grid sampling [33] which results in equal point distance. The voxel size is tuned to give approximately 1000 seed points per. object in both the query and target. The target seed points are found by transforming the query seeds into the target by applying the object ground truth pose. The target seeds are selected in a nearest neighbour search with a distance threshold. A feature descriptor for each seed point in the query and target mesh is computed. For a fair comparison individually tuned support radii for each descriptors are used. We use the following feature resolution multiplier: SI(20), 3DSC(22.5), FPFH(17.5), USC(25), SHOT(17.5), ROPS(20), ECSAD(20), NDHIST(31). Hence, the radius for each feature is a function of the average model or scene resolution. Upon feature computation the underlying scene and object meshes are utilized in 0.25 and 0.05 decimated version; respectively. The level of decimation are empirical determined to the level with best overall matching results for all features. During decimation the normal orientation is re-computed for each vertex by the area weighted mean of the mesh triangle [40] and normalized. In order to resolve the exact number of correct feature matches, a brute-force linear kd-tree search is used with a L_2 distance function. Other distance metrics such as L_1, L_∞, are tested in previous work but the best results are achieved with the L_2 distance metric. Thus, we only present results for the L_2 metric. During matching we are computing the ratio of the nearest and the second-nearest matching distances. This matching strategy is adapted from multiple previous studies e.g. Lowe *et al*. [26] that proved a performance enhancement compared to a native matching strategy where only the nearest neighbour is considered. Once all matches for all queries are computed they are ranked and sorted according to the L_2 distance in one array. The correct matches are found by traversing the array of matches and count the number of matches that are spatial close, determined by a distance threshold. The PRC curves are presented in Section 5 where precision refers to the number of correct matches compared to the total number of matches. Recall refers to the number of correct matches compared to the total amount of possible matches (i.e. feature seed points found in the target). In addition to PRC curves, we compute the area under the PRC curve (AUC) as a single quantitative measure of the overall accuracy. The AUC is computed as the numerical integration over all (P,R) per feature.

4.2. Pose estimation

In this section the experimental protocol for the pose estimation experiments is presented. The sampling and seed point selection are identical with the feature matching benchmark presented, except that we cannot use the ground truth pose for selecting target seed points. Instead, the target resolution is doubled, thus quadrupling the number of feature descriptors which increase the chance for describing the same feature. To increase the efficiency during

matching we apply approximate nearest neighbour search to determine correspondences hypotheses instead of exact matching. Again, the ratio of the nearest and second nearest neighbour feature distances are used. A multiple randomized kd-trees with a bound of 512 checks and 4 trees are used as a good trade-off between accuracy and efficiency. Correspondences are ranked by the L_2 distance which inputs potential feature correspondences to a hypothesis and test RANSAC algorithm. During random sampling, three correspondences are sampled which is sufficient to generate a hypothesis pose. The hypothesis pose is tested by transforming the query points and counting the number of query points close to the target feature up to a tolerance given by the inlier threshold. The algorithm filters out false positives by setting a lower minimum of the number of inliers required to accept a pose hypothesis to 1%. The pose with the highest number of inliers is returned as the object pose. Our RANSAC implementation deviates from classic RANSAC, which treats all data points uniformly. Instead we sample correspondences according to their quality score. The quality scores are given by the negative normalized L_2 distance ratio. The efficiency of the algorithm is further increased by only considering the top 10% of the best correspondences with the highest quality score before running the RANSAC algorithm. Upon RANSAC completion the final pose is refined by 150 ICP iterations on the query/target seed points. We accept a pose estimate as valid by computing the euclidean and geodesic distances between the computed pose and the ground true pose from the annotation process. If the euclidean and geodesic distances are less than the threshold the object is correctly recognized. The euclidean and geodesic distance metrics are computed in accordance to Equation 4 and 5.

$$arccos\left(\frac{trace(\mathbf{R}^T\hat{\mathbf{R}})-1}{2}\right) \leq 7.5^\circ \tag{4}$$

$$||\mathbf{t}-\hat{\mathbf{t}}|| \leq 5mm \tag{5}$$

The recognition rate is computed as the ratio of true positive poses compared to all detected poses as a quantitative measure of the overall recognition performance.

5. Evaluation

In this section we present the results of our evaluation benchmark. All experiments are based on the proposed dataset where 3204 views, 45 objects and 5675 ground truth poses are included. As a first evaluation we benchmark the matching accuracy of the seven feature descriptors with the parameters outlined in Section 4.1 and all 45 object models included. The PRC curve of the overall matching accuracy is presented in Figure 4. As expected the total matching accuracy is much lower compared to previous studies e.g. Guo *et al.* [17] and Buch *et al.* [4], where both are running a similar benchmark but on previous proposed datasets. The result indicates that our objective of the proposed dataset has been fulfilled. Moreover, the evaluation shows that the ECSAD feature has a best performance with a AUC = 0.046 followed by ROPS with AUC = 0.011. In order to investigate the performance further we run the matching benchmark for each object model and compute PRC curves for each of the 45 models. We have categorize the object into three different groups and present 3 PRC curves for each group in Figure 7-14 as a sample. The three groups are a) Geometric complex objects, b) Cylindrical objects and c) Flat or box shaped objects. The objects that corresponds to the PRC curves in Figure 7-14 are presented in Figure 6. The geometric complex objects are the Angel, the Birds and the Rabbit in Figure 6(a)-(c), the cylindrical objects are Neutral, Pringles and Hand soap in Figure 6(d)-(f), and the flat or box shaped objects are the button, the brake disc and the Psu in Figure 6(g)-(i). The results show that a recent precision/recall is achieved with the Angel,Birds and Rabbit, but it is much lower than previous studies e.g. Buch *et al.* [4] who obtain very matching accuracy for some datasets. These low numbers for our geometric ideal objects indicate that the our dataset has the desired level of complexity. Regarding, the cylindrical objects which features the Neutral, Pringles and Hand Soap objects, it is clear that current local shape descriptors are very little descriptive for these uniform shaped objects. Again, ECSAD performs in general best which might results from ECSAD' ability to capture edges. For the flat and boxed shaped models we can conclude that current state of the art feature descriptors is not suitable as the only detection method.

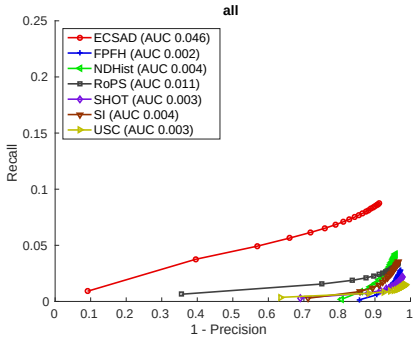
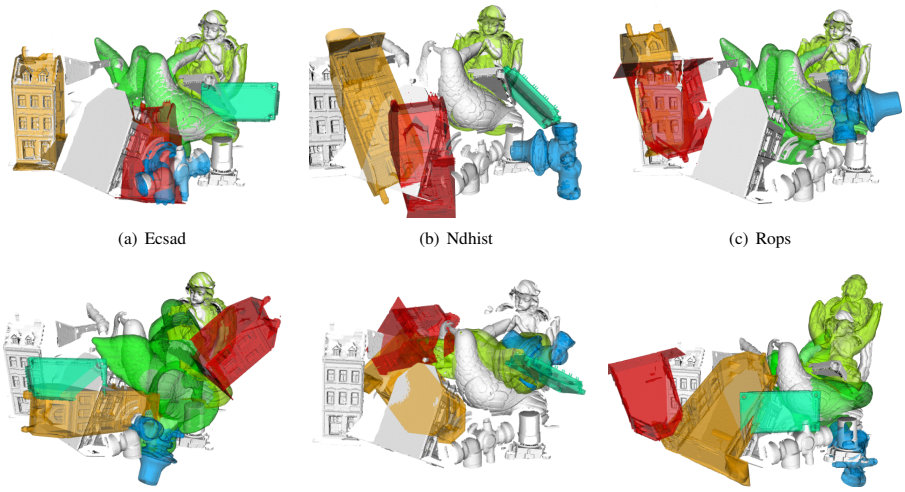


Figure 4: Overall PRC curve

The results for the pose estimation experiments are presented in Table 2 and Figure 5. Our object recognition pipeline in these experiment is in accordance to the presented pipeline in Section 4.2. In the first recognition exper-

Feature	Overall	Angel	Birds	Rabbit	Neutral	Pringels	Hand soap	Button	Brake Disc	Psu	*Mean* (λ)
ECSAD	0.21	**0.59**	**0.86**	**0.62**	0.00	0.06	**0.03**	0.00	0.00	0.00	**0.24**
FPFH	0.01	0.07	0.01	0.04	0.00	0.00	0.02	0.00	0.00	0.00	0.02
NDHIST	0.02	0.13	0.05	0.14	**0.02**	**0.06**	0.02	0.00	0.00	0.00	0.05
ROPS	0.13	0.45	0.70	0.29	0.01	0.00	0.00	0.00	0.00	0.00	0.16
SHOT	0.04	0.25	0.09	0.14	0.00	0.00	0.00	0.00	0.00	0.00	0.05
SI	0.05	0.20	0.16	0.10	0.01	0.00	0.00	0.00	0.00	0.00	0.05
USC	0.00	0.01	0.04	0.09	0.00	0.00	0.00	0.00	0.00	0.00	0.02

Table 2: Overall recognition rates. **Column 1:** Features descriptors. **Column 2:** Overall recognition rate. **Column 3-11:** Recognition rate for each sample object in Figure 6. **Column 12:** Mean recognition rate for the 9 sample object per. feature.



(a) Ecsad (b) Ndhist (c) Rops

(d) Shot (e) Fpfh (f) Spin images

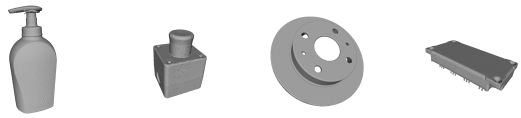
Figure 5: Qualitative recognition results for view 94

iment we run the pipeline with all 45 object model included and present the recognition rate in Table 2, second column. In the second experiment we run the recognition benchmark for each individual object, which is a more complex problem because we in the benchmark remove the scene points corresponding to each detected object. Hence, in case of more objects to recognize, the number of scene points are reduced each time the algorithm detect an object. Again, as expected ECSAD and ROPS perform best on average. However, the recognition rate is lower than seen in other datasets which is expected since some views in our dataset are heavy occluded. In Figure 5 qualitative results are presented for the recognition result in view 94. Again, is is clear that ECSAD and ROPS perform best with 4 and 2 correct recognize objects, respectively.

6. Conclusion

This paper has introduce a new large scale dataset for 3D object recognition and a evaluation benchmark to validate the dataset. Our benchmark results show as expected a general low matching score due to the level of complexity of the dataset. Especially, repetitive, symmetric, flat and thin-edge objects without many local features demonstrate as expected a very low precision/recall. Our matching results shows that the ECSAD feature performs best followed by the ROPS feature. This result is in accordance to some of the experiment in a previous benchmark by Buch *et al*. [4]. Our object recognition results reflected as expected the low matching accuracy from the matching experiments. From the evaluation we conclude that the dataset full-fill our requirement in terms of difficulty. Furthermore, the objects and scenes included in the dataset represents the real world problems, which 3D object recognition systems need to handle in the future. Our main goal of this work has been to challenge existing 3D object recognition algorithms and create a dataset which contains real world object from the robot and automation industry.

(a) Angel (b) Birds (c) Rabbit (d) Neutral (e) Pringels



(f) Hand soap (g) Button (h) Brake disc (i) Psu

Figure 6: (a)-(c) Geometric complex objects used for dataset verification. (d)-(e) Cylinderical objects. (f)-(h) Flat and box shaped objects

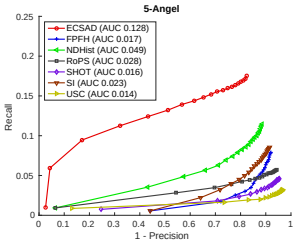


Figure 7: PRC curve: Angel

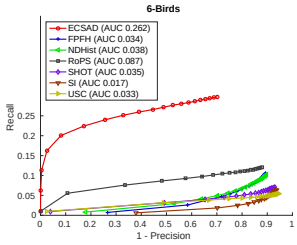


Figure 8: PRC curve: Birds

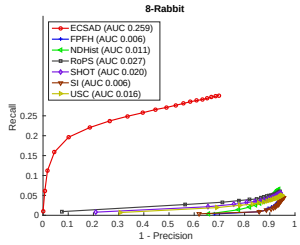


Figure 9: PRC curve: Rabbit

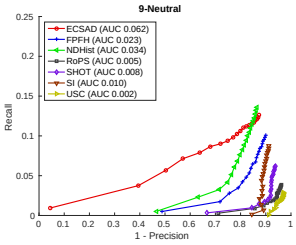


Figure 10: PRC curve: Neutral

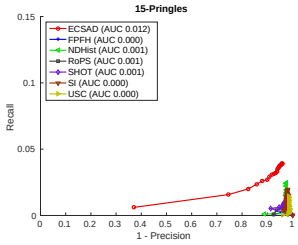


Figure 11: PRC curve: Pringles

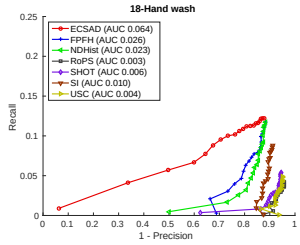


Figure 12: PRC curve: Hand soap

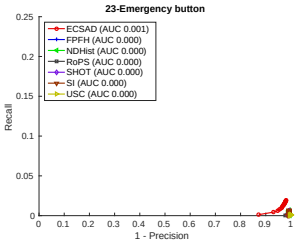


Figure 13: PRC curve: Button

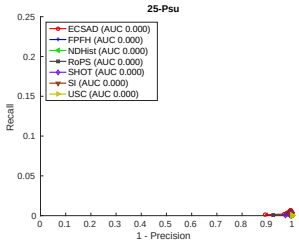


Figure 14: PRC curve: Psu

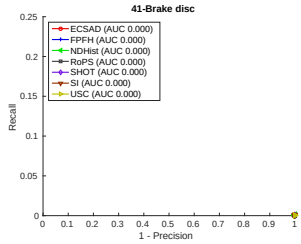


Figure 15: PRC curve: Brake disc

References

[1] H. Aanæs, A. L. Dahl, and K. S. Pedersen. Interesting interest points - A comparative study of interest point performance on a unique data set. *International Journal of Computer Vision*, 97(1):18–35, 2012.

[2] A. Aldoma, F. Tombari, L. Di Stefano, and M. Vincze. A global hypotheses verification method for 3d object recognition. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part III*, ECCV'12, pages 511–524, Berlin, Heidelberg, 2012. Springer-Verlag.

[3] T. Birdal and S. Ilic. Point pair features based object detection and pose estimation revisited. In *2015 International Conference on 3D Vision, 3DV 2015, Lyon, France, October 19-22, 2015*, pages 527–535, 2015.

[4] A. G. Buch, H. G. Petersen, and N. Krüger. Local shape feature fusion for improved matching, pose estimation and 3d object recognition. *SpringerPlus*, 5(1):1–33, 2016.

[5] H. Chen and B. Bhanu. 3d free-form object recognition in range images using local surface patches. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 136–139 Vol.3, Aug 2004.

[6] M. Corsini, P. Cignoni, and R. Scopigno. Efficient and flexible sampling with blue noise properties of triangular meshes. *IEEE Transactions on Visualization and Computer Graphics*, 18(6):914–924, June 2012.

[7] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255, June 2009.

[8] F. Dornaika and R. Horaud. Simultaneous robot-world and hand-eye calibration. *IEEE Transactions on Robotics and Automation*, 14(4):617–622, Aug 1998.

[9] E. Eiriksson, J. Wilm, D. Pedersen, and H. Aans. Precision and accuracy parameters in structured light 3-d scanning. In *Proceedings of LowCost3D: Sensors, Algorithms, Application (2015).*, pages 7–15, 2016.

[10] M. Everingham, S. M. A. Eslami, L. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2014.

[11] S. Filipe and L. A. Alexandre. A comparative evaluation of 3d keypoint detectors in a RGB-D object dataset. In *VISAPP 2014 - Proceedings of the 9th International Conference on Computer Vision Theory and Applications, Volume 1, Lisbon, Portugal, 5-8 January, 2014*, pages 476–483, 2014.

[12] A. Flint, A. Dick, and A. v. d. Hengel. Thrift: Local 3d structure recognition. In *Digital Image Computing Techniques and Applications, 9th Biennial Conference of the Australian Pattern Recognition Society on*, pages 182–188, Dec 2007.

[13] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik. *Computer Vision - ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part III*, chapter Recognizing Objects in Range Data Using Regional Point Descriptors, pages 224–237. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[14] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, J. Garcia-Rodriguez, J. Azorin-Lopez, M. Saval-Calvo, and M. Cazorla. Multi-sensor 3d object dataset for object recognition with full pose estimation. *Neural Computing and Applications*, pages 1–12, 2016.

[15] J. Guehring. Dense 3d surface acquisition by structured light using off-the-shelf components. volume 4309, pages 220–231, 2000.

[16] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, and J. Wan. 3d object recognition in cluttered scenes with local surface features: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2270–2287, Nov 2014.

[17] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N. M. Kwok. A comprehensive performance evaluation of 3d local feature descriptors. *International Journal of Computer Vision*, 116(1):66–89, 2015.

[18] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, and J. Wan. Rotational projection statistics for 3d local surface description and object recognition. *International Journal of Computer Vision*, 105(1):63–86, 2013.

[19] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, and V. Lepetit. Gradient response maps for real-time detection of textureless objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):876–888, May 2012.

[20] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(5):433–449, May 1999.

[21] T. B. Jørgensen, A. G. Buch, and D. Kraft. *Geometric Edge Description and Classification in Point Cloud Data with Application to 3D Object Recognition*. 2015.

[22] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, SGP '06, pages 61–70, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.

[23] K. Lai, L. Bo, and D. Fox. Unsupervised feature learning for 3d scene labeling. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3050–3057, May 2014.

[24] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1817–1824, May 2011.

[25] K. Lai, L. Bo, X. Ren, and D. Fox. Detection-based object labeling in 3d scenes. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1330–1337, May 2012.

[26] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, Nov. 2004.

[27] A. Mian, M. Bennamoun, and R. Owens. Three-dimensional model-based object recognition and segmentation in cluttered scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(10):1584–1601, Oct 2006.

[28] A. Mian, M. Bennamoun, and R. Owens. On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. *International Journal of Computer Vision*, 89(2-3):348–361, 2010.

[29] C. Rennie, R. Shome, K. E. Bekris, and A. F. D. Souza. A dataset for improved rgbd-based object detection and pose estimation for warehouse pick-and-place. *CoRR*, abs/1509.01277, 2015.

[30] E. Rodol, A. Albarelli, F. Bergamasco, and A. Torsello. A scale independent selection process for 3d object recognition in cluttered scenes. *International Journal of Computer Vision*, 102(1-3):129–145, 2013.

[31] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 3212–3217, May 2009.

[32] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz. Aligning point cloud views using persistent feature histograms. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3384–3391, Sept 2008.

[33] R. B. Rusu and S. Cousins. 3d is here: Point cloud library (pcl). In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1–4, May 2011.

[34] S. Salti, F. Tombari, and L. D. Stefano. A performance evaluation of 3d keypoint detectors. In *2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pages 236–243, May 2011.

[35] S. Salti, F. Tombari, and L. D. Stefano. Shot: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125(0):251 – 264, 2014.

[36] A. Singh, J. Sha, K. Narayan, T. Achim, and P. Abbeel. Bigbird: A large-scale 3d database of object instances. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 509–516, May 2014.

[37] F. Stein and G. Medioni. Structural indexing: efficient 3-d object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):125–145, Feb 1992.

[38] B. Taati, M. Bondy, P. Jasiobedzki, and M. Greenspan. Variable dimensional local shape descriptors for object recognition in range data. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct 2007.

[39] B. Taati and M. Greenspan. Local shape descriptor selection for object recognition in range data. *Computer Vision and Image Understanding*, 115(5):681 – 694, 2011. Special issue on 3D Imaging and Modelling.

[40] G. Thürmer and C. A. Wüthrich. Computing vertex normals from polygonal facets. *J. Graph. Tools*, 3(1):43–46, Mar. 1998.

[41] F. Tombari, S. Salti, and L. Di Stefano. Unique shape context for 3d data description. In *Proceedings of the ACM Workshop on 3D Object Retrieval*, 3DOR '10, pages 57–62, New York, NY, USA, 2010. ACM.

[42] F. Tombari, S. Salti, and L. Di Stefano. Unique signatures of histograms for local surface description. In *Proceedings of the 11th European Conference on Computer Vision Conference on Computer Vision: Part III*, ECCV'10, pages 356–369, Berlin, Heidelberg, 2010. Springer-Verlag.

[43] W. G. . T. U. Wien. The willow garage object recognition challenge, 2015. [Online; accessed 5-May-2015].

[44] A. Zaharescu, E. Boyer, K. Varanasi, and R. P. Horaud. Surface feature detection and description with applications to mesh matching. In *International Conference on Computer Vision and Pattern Recognition, CVPR'09, June, 2009*, pages 373–380, Miami, Etats-Unis, June 2009. IEEE.

[45] Y. Zhong. Intrinsic shape signatures: A shape descriptor for 3d object recognition. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 689–696, Sept 2009.

4.5 Discussion and conclusion

In this chapter two contributions are presented. The first contribution, [Contribution E] demonstrates a vision system for fast 3D modelling of larger objects. With the vision system a production worker is able to teach a robot to localize novel objects by placing the object in the scene. When a full 3D model is obtained, the model is used for 3D pose estimation. The experimental results show that the created model is good enough as model in a 3D pose estimation pipeline.

The other contribution [Contribution F] presents a new large-scale dataset for evaluation of 3D pose estimation algorithm. The dataset is significantly larger in size than any other proposed 3D pose estimation dataset. The evaluation benchmark evaluates 7 state of the art local shape features on the data. The experimental results show that ECSAD is curently the best local shape feature. Furthermore, the results show as expected that current proposed local shape features are not descriptive enough to properly detect flat and cylindrical objects in presence of large amount of clutter and occlusion. In order to improve the benchmark results new novel shape features, which are better to describe the surface are needed.

Chapter 5

Conclusion and future work

A number of contributions to the problem domain of robot guidance have been presented in this thesis. The work presented in Chapter 2 emphasized the problems and challenges in robot guidance by 2D robot vision. In the chapter a vision system and a framework, which enable easy instruction of an industrial robot through a skill based framework are presented. The vision system implemented algorithms for 2D, 2.5D and 3D single camera pose estimation. This work is published in [Contribution B] and [Contribution C]. During, the development of the system, experiences illustrated the difficulties in making a smooth vision system, which production workers are able to fast re-configure. The main challenge is to develop a smooth calibration flow of the system in 2D applications. Moreover, the training process of new patterns and assignment of world models with object feature coordinates is difficult to make intuitive for a production worker. In 2.5D and 3D pose estimation applications where world models are requires, knowledge in vision systems is crucial to make the application running properly.

The experience from the work in [Contribution B] and [Contribution C] is that vision systems for robot guidance need to be more intelligent and learn perception models. This experience triggered the work presented in [Contribution E] where a robot learns a 3D perception model. However, the physical camera setup in this work is comprehensive but with this contribution we showed that a simple and noisy 3D model representation is accurate enough to estimate the

pose of larger objects. In the future work, hopefully we will see commercial robots utilizing these ideas. Either, approaches where a robot grasps objects in order to rotate it in front of camera or approaches where the robot moves around object to explore it. These methods have already been proposed in robotic research e.g. [KHRF11], [ALRC14].

Many of the proposed systems for learning perception models are in full 3D e.g. [KHRF11], [ALRC14] and [Contribution E]. However, until the general performance and speed of 3D pose estimation algorithms increase and more low cost 3D sensors enter the marked, learning methods in 2D are required. In humanoid robotic research this is a topic, which has gained much interest. Within this research area perception models are build by accumulating local or global point or edge features. Examples of this research include [WIS+10],[BU15] and the presented related work in Section 2.7. Unfortunately, the focus is still textured objects, which are not applicable in industrial robotics. Recently, the Canadian company Robotiq introduced a commercial system for a Universal Robot which is based on these ideas [1]. Their system consists of a small mobile camera, mounted at the flange of the robot and a vision system integrated directly in the robot controller. They learned the object representation by taking images from different views and sore the model. The accuracy of the model and the performance of the system in presence of occlusion and clutter is unknown but the fact that the first commercial product has entered the marked shows that we are going in the right direction.

In chapter 3 methods for 3D estimation are presented and discussed. In general, cost-efficient and accurate commercial 3D sensors suited for industrial automation are still not available on the marked. This is a prerequisite for reliable 3D pose estimation applications. A general commercial 3D picking solution where robots learn perception model of e.g. the industrial objects presented in Figure 1.5 and afterwards are able to estimate an accurate pose, is not possible before better, smaller and cost-efficient 3D sensor enter the marked. In [Contribution D] a structured light sensor, which fulfil these requirements are presented. In this work a new novel structured light sensor, which includes some of the features needed in order to estimate the object surface of many of the object presented in Figure 1.5. The size of the sensor is small enough to be mounted in the tool of a collaborative robot. With this sensor and current state of the art pose estimation algorithms it is possible to develop reliable box picking applications where objects with many geometric features are detected.

However, for objects with simpler geometry like the once presented in Figure 1.5, current state of the art local shape feature is not descriptive enough. This is proven in [Contribution F] where a new large-scale dataset is proposed. The

[1] `http://robotiq.com/products/camera/`

dataset consists of 3204 scene views and 45 object models, which is the largest dataset for 3D pose estimation until today. The object models included in the dataset consist of flat and cylindrical objects with many shape features. The evaluation benchmark presented in [contribution F] point out that local shape features are not descriptive enough to detect flat and cylindrical objects. We conclude that planar objects and objects with/without texture, which primarily consists of 3D edges or corner information, but little shape variation is not possible to detect with current state of the art local shape features. In order to detect planar objects novel methods that combine edge information and point pair relations are required. As shown in [contribution F] the best descriptor is the ECSAD descriptor which is originally developed for extracting edges from point clouds. The fact that ECSAD is the best performing descriptor tells us that new local feature descriptors could benefit from using point cloud edges. Previous studies like Drost *et al.* [DUNI10] presented good experimental results with features which use the relation between local feature descriptors. This approach combined with edge descriptors could be a future solution for increasing the matching performance for objects with few geometric features. Even combining three local feature points in triplets features could increase the performance. All the mentioned suggestions rely on hand-crafted feature descriptors which have shown to be difficult to generalize. Previous work in 2D object recognition has presented a significant increase in the detection performance after introducing feature learning methods like e.g. convolutional neural networks to learn the best feature vector for a given object. This approach could help increasing the performance in 3D.

The overall goal of this thesis is to investigate how a 3D object detection pipeline, which is able to detect and estimate the pose of a large variety of objects is constructed. Thus, by providing a robot with a CAD model or the capability to learn the object representation the robot is able to detect the required objects. Based on the experience and scientific results during the project advances are required within two areas. First, better in-expensive commercial 3D sensors which are resistant towards inter-reflections and sub-surface scattering with better dynamic range are required. Algorithms to make 3D sensors resistant towards these phenomenon are developed by the research community but the algorithms must be included in in-expensive commercial 3D sensors before 3D picking applications is cost-efficient compared to an engineered 2D vision solution. Secondly, better 3D features which are able to describe simple surfaces are required. When this is realised, a cost-efficient 3D Plug-n-Play robotic guidance system could be more attractive in terms of generality, compared to an engineered 2D vision solution.

Appendix A

Contribution A

This appendix contains the **Contribution A** paper published at the 2nd AAU Workshop on Robotics held in Aalborg, Denmark the 30. October 2013 at Aalborg University. The one page abstract and the reference to the paper are following below.

Thomas Sølund, Rasmus Hasle Andersen, Anders Billesø Beck, and Henrik Aanæs. Combining 3D Object Modelling and Robot Skills for Intuitive Instruction of Robotic co-workers. In *2nd AAU Workshop on Robotics*, 2013. Peer-reviewed

Combining 3D Object Modelling and Robot Skills for Intuitive Instruction of Robotic co-workers

Thomas Sølund, Rasmus Hasle Andersen & Anders B. Beck
Danish Technological Institute
Robot Technology
DK-5230 Odense M
Email: [thso, raha, anbb]@dti.dk

Henrik Aanæs
Technical University of Denmark
Department of Applied Mathematics and Computer Science
DK-2800 Kgs. Lyngby
Email: aanes@dtu.dk

***Abstract*—Automatic extraction of CAD models which can serve as input for robot manipulation and object recognition algorithms requires the accurate inference of geometric and topological information. We present our ongoing work on creating a system which applies structured light techniques to automatically extract CAD models of parts with an industrial robot. Our aim is to create a 3D modelling system that enables a human to create new CAD models directly on site in the production by instructing a robot to move around the object of interest and create new models. This is realized through an intuitive robot skill framework.**

I. Introduction

It is well known that the Danish production industry is under pressure due to high wage and staff expenses. Automation can be a solution to lower these expenses. A task which is difficult when robots need flexibility to cope for the high rate of change in a modern production. Flexibility which is essential in small and medium enterprises (SMEs) to make robots profitable.
Combining skill based robot programming and a scanning tool to infer new 3D object models creates a strong framework which allows fast adapting to new tasks and changes in a robot cell. Additional, modelling capabilities give the flexibility to create scene models that can provide motion planners and obstacle avoidance systems with geometric data of a robot cell. This reduces the amount of manual work required in changing geometric scene descriptions when a new cell configuration is required in a production. Furthermore, the object models can provide a suitable model for a geometric grasp planner.

Inferring consistent 3D models from range scans was introduced over two decades ago [1]. Combining reconstruction of free-form object and a robot for changing view point has a growing potential in both the domestic [2] and industrial robot [3] domain. Bone et.al. used photometric imaging to make a rough reconstruction of a nut and afterwards refine the model with a laser strip scanner. Our work builds on a structured light scanner which directly gives us shape information without the need of computing a prior shape. A modelling framework based on a RGD-B camera was successful shown by Krainin et. al. They developed an ICP based algorithm which continually track the object when a robot manipulator rotate the object in front of the camera.

II. 3D Modelling framework

Our 3D surface reconstruction is based on a stereo vision system and a structured light projector. By projecting a temporal gray coded pattern onto the surface and observe the deflection of the vertical edges, we are able to create a left and right encoded image. Searching the epipolar line for similarly coded pixels gives a disparity which allows the depth to be reconstructed using linear triangulation.
Each scanning position is entirely controlled by the user which creates a new skill through our graphical user interface in figure 1. By manually moving the robot to different scanning positions and parametrize the skill the user is able to create a robot program which moves the sensor around the object of interest. We plan to extend this process by doing semi-autonomous view planning, by allowing the user to give 4-5 fixed positions which create a rough model. The rough model is then used to plan the necessary views with a Next Best View planner. Each scan is registered using the iterative closest point(ICP) algorithm to create a full model of the object. As an initial alignment of each scan before ICP registration we use the current pre-calibrated view point from the robot.

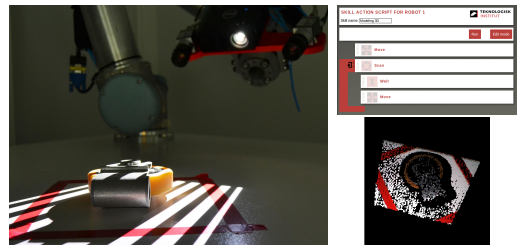


Fig. 1. **Left:** DTI Robot co-worker platform for 3D scanning **Upper Right:** Graphical user interface to construct a new skill by combining hardware dependent primitives. **Lower Right:** Output of our 3D scanning system

References

[1] Y. Chen and G. Medioni, "Object Modeling by Registration of Multiple Range Images," in *Proceedings of the 1991 IEEE International Conference on Robotics and Automation Sacramento, California*, no. April, 1991, pp. 2724–2729.

[2] M. Krainin, P. Henry, X. Ren, and D. Fox, "Manipulator and object tracking for in-hand 3D object modeling," *The International Journal of Robotics Research*, vol. 30, no. 11, pp. 1311–1327, Jul. 2011. [Online]. Available: http://ijr.sagepub.com/cgi/doi/10.1177/0278364911403178

[3] G. M. Bone, A. Lambert, and M. Edwards, "Automated modeling and robotic grasping of unknown three-dimensional objects," in *2008 IEEE International Conference on Robotics and Automation*. Ieee, May 2008, pp. 292–298. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4543223

Bibliography

[AAD07] Pedram Azad, Tamim Asfour, and R. Dillmann. Stereo-based 6d object localization for grasping with humanoid robot systems. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 919–924, Oct 2007.

[AAD09] Pedram Azad, Tamim Asfour, and Rüdiger Dillmann. *Autonome Mobile Systeme 2009: 21. Fachgespräch Karlsruhe, 3./4. Dezember 2009*, chapter Stereo-Based vs. Monocular 6-DoF Pose Estimation Using Point Features: A Quantitative Comparison, pages 41–48. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[AAK71] Y.I. Abdel-Aziz and H.M. Karara. *Direct Linear Transformation from Comparator Coordinates Into Object Space Coordinates in Close-range Photogrammetry.* 1971.

[ABBP07] J. Assfalg, M. Bertini, A. Del Bimbo, and P. Pala. Content-based retrieval of 3-d objects using spin image signatures. *IEEE Transactions on Multimedia*, 9(3):589–599, April 2007.

[ABD12] Pablo Fernández Alcantarilla, Adrien Bartoli, and Andrew J. Davison. *KAZE Features*, pages 214–227. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[AC95] M. A. Abidi and T. Chandra. A new efficient and direct solution for pose estimation using quadrangular targets: algorithm and evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):534–538, May 1995.

[AD03] A. Ansar and K. Daniilidis. Linear pose estimation from points or lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):578–589, May 2003.

[ADSP12] Henrik Aanæs, Anders Lindbjerg Dahl, and Kim Steenstrup Pedersen. Interesting interest points. *Int. J. Comput. Vision*, 97(1):18–35, March 2012.

[AHB87] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):698–700, Sept 1987.

[AKB08] Motilal Agrawal, Kurt Konolige, and Morten Rufus Blas. *Computer Vision – ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part IV*, chapter CenSurE: Center Surround Extremas for Realtime Feature Detection and Matching, pages 102–115. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[ALRC14] Jacopo Aleotti, Dario Lodi Rizzini, and Stefano Caselli. Perception and grasping of object parts from active robot exploration. *Journal of Intelligent & Robotic Systems*, 76(3):401–425, 2014.

[AMN+98] Sunil Arya, David M. Mount, Nathan S. Netanyahu, Ruth Silverman, and Angela Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *J. ACM*, 45(6):891–923, November 1998.

[And15] Rasmus Hasle Andersen. *Towards Cost-Effective Robotic Systems for Small and Medium Enterprises.* PhD thesis, 2015.

[AOV12] A. Alahi, R. Ortiz, and P. Vandergheynst. Freak: Fast retina keypoint. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 510–517, June 2012.

[ASH13] Rasmus Hasle Andersen, Thomas Sølund, and John Hallam. Definition of Hardware-Independent Robot Skills for Industrial Robotic Co-workers. In *IEEE/RSJ International Conference on Intelligent Robots and Systems 2013 - Workshop on Robotic Assistance Technologies in Industrial Settings (RATIS)*, pages 1–7, Tokyo, Japan, 2013. Peer-reviewed.

[ASH14] Rasmus Hasle Andersen, Thomas Sølund, and John Hallam. Definition and Initial Case-Based Evaluation of Hardware-Independent Robot Skills for Industrial Robotic Co-Workers. In *Proceedings of 41st International Symposium on Robotics (ISR/Robotik 2014)*, pages 101–107, 2014. Peer-reviewed.

[AT13] Alexander Andreopoulos and John K. Tsotsos. 50 years of object recognition: Directions forward. *Computer Vision and Image Understanding*, 117(8):827 – 891, 2013.

[ATQ00] Marc-André Ameller, Bill Triggs, and Long Quan. Camera Pose Revisited – New Linear Algorithms. 2000. Submitted to ECCV'00.

[BBBB+15] Michael Beetz, Ferenc Balint-Benczedi, Nico Blodow, Daniel Nyga, Thiemo Wiedemeyer, and Zoltan-Csaba Marton. RoboSherlock: Unstructured Information Processing for Robot Perception. In *IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, Washington, USA, 2015. Best Service Robotics Paper Award.

[BBH03] M. Z. Brown, D. Burschka, and G. D. Hager. Advances in computational stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):993–1008, Aug 2003.

[Bea78] P. R. Beaudet. Rotationally invariant image operators. In *International Conference on Pattern Recognition*, 1978.

[Ben75] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, September 1975.

[BIH+12] D. Alex Butler, Shahram Izadi, Otmar Hilliges, David Molyneaux, Steve Hodges, and David Kim. Shake'n'sense: Reducing interference for overlapping structured light depth cameras. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1933–1936, New York, NY, USA, 2012. ACM.

[BKL06] Alina Beygelzimer, Sham Kakade, and John Langford. Cover trees for nearest neighbor. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 97–104, New York, NY, USA, 2006. ACM.

[BKZ14] Tyler Bell, Nikolaus Karpinsky, and Song Zhang. *Real-Time 3D Sensing With Structured Light Techniques*, pages 181–213. John Wiley Sons, Ltd, 2014.

[BL97] Jeffrey S. Beis and David G. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, CVPR '97, pages 1000–, Washington, DC, USA, 1997. IEEE Computer Society.

[Bla04] François Blais. Review of 20 years of range sensor development. *J. Electronic Imaging*, 13(1):231–243, 2004.

[BM92] PJ Besl and ND McKay. A method for registration of 3-D shapes. *IEEE Transactions on pattern analysis and machine ...*, 1992.

[BMP02] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, Apr 2002.

[BMS98] J Batlle, E Mouaddib, and J Salvi. Recent progress in coded structured light as a technique to solve the correspondence problem: a survey. *Pattern recognition*, 31(7), 1998.

[BPK16] Anders G. Buch, Henrik G. Petersen, and Norbert Krüger. Local shape feature fusion for improved matching, pose estimation and 3d object recognition. *SpringerPlus*, 5(1):1–33, 2016.

[Bri95] Sergey Brin. Near neighbor search in large metric spaces. In *Proceedings of the 21th International Conference on Very Large Data Bases*, VLDB '95, pages 574–584, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[BRS+11] Kai Berger, Kai Ruhl, Yannic Schroeder, Christian Bruemmer, Alexander Scholz, and Marcus Magnor. Markerless Motion Capture using multiple Color-Depth Sensors. In Peter Eisert, Joachim Hornegger, and Konrad Polthier, editors, *Vision, Modeling, and Visualization (2011)*. The Eurographics Association, 2011.

[BTVG06] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. *Computer Vision – ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I*, chapter SURF: Speeded Up Robust Features, pages 404–417. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[BU15] R. Bevec and A. Ude. Pushing and grasping for autonomous learning of object models with foveated vision. In *Advanced Robotics (ICAR), 2015 International Conference on*, pages 237–243, July 2015.

[CB04] Hui Chen and Bir Bhanu. 3d free-form object recognition in range images using local surface patches. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 136–139 Vol.3, Aug 2004.

[CBK04] German K. M. Cheung, Simon Baker, and Takeo Kanade. Shape-from-silhouette across time part I: theory and algorithms. *International Journal of Computer Vision*, 62(3):221–247, 2004.

[CBSF09] Alvaro Collet Romea, Dmitry Berenson, Siddhartha Srinivasa, and David Ferguson . Object recognition and full pose registration from a single image for robotic manipulation. In *IEEE International Conference on Robotics and Automation (ICRA '09)*, May 2009.

[CC12] C. Choi and H. I. Christensen. 3d textureless object detection and tracking: An edge-based approach. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3877–3884, Oct 2012.

[CKS98] D. Caspi, N. Kiryati, and J. Shamir. Range imaging with adaptive color structured light. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):470–480, May 1998.

[CL95] Brian Curless and Marc Levoy. *Better optical triangulation through spacetime analysis*, pages 987–994. IEEE, 1995.

[CL96] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques - SIGGRAPH '96*, pages 303–312, New York, New York, USA, 1996. ACM Press.

[CLSF10] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. *Computer Vision – ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV*, chapter BRIEF: Binary Robust Independent Elementary Features, pages 778–792. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[CLV12] Leandro Cruz, Djalma Lucio, and Luiz Velho. Kinect and rgbd images: Challenges and applications. In *SIBGRAPI Tutorial*, 2012.

[CMS11] Alvaro Collet Romea, Manuel Martinez Torres, and Siddhartha Srinivasa. The moped framework: Object recognition and pose estimation for manipulation. *International Journal of Robotics Research*, 30(10):1284 – 1306, September 2011.

[CS10] Alvaro Collet Romea and Siddhartha Srinivasa. Efficient multi-view object recognition and full pose estimation. In *2010 IEEE International Conference on Robotics and Automation (ICRA 2010)*, May 2010.

[CSF12] M. M. Bronstein C. Strecha, A. M. Bronstein and Pascal Fua. LDAHash: Improved Matching with Smaller Descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1), 2012.

[Dan98] Konstantinos Daniilidis. Hand-eye calibration using dual quaternions. *International Journal of Robotics Research*, 18:286–298, 1998.

[DAP11] Anders Lindbjerg Dahl, Henrik Aanæs, and Kim Steenstrup Pedersen. Finding the best feature detector-descriptor combination. In *Proceedings of the 2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, 3DIMPVT '11, pages 318–325, Washington, DC, USA, 2011. IEEE Computer Society.

[DD95] Daniel F. Dementhon and Larry S. Davis. Model-based object pose in 25 lines of code. *Int. J. Comput. Vision*, 15(1-2):123–141, June 1995.

[DH98] F. Dornaika and R. Horaud. Simultaneous robot-world and hand-eye calibration. *IEEE Transactions on Robotics and Automation*, 14(4):617–622, Aug 1998.

[DK06] H. Q. Dinh and S. Kropac. Multi-resolution spin-images. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 863–870, June 2006.

[DK12] T. Darom and Y. Keller. Scale-invariant features for 3-d mesh models. *IEEE Transactions on Image Processing*, 21(5):2758–2769, May 2012.

[DNRR05] James Davis, Diego Nehab, Ravi Ramamoorthi, and Szymon Rusinkiewicz. Spacetime Stereo: A unifying framework for depth from triangulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(2):296–302, February 2005.

[DPP09] R. Detry, N. Pugeault, and J. H. Piater. A probabilistic framework for 3d visual object representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1790–1803, Oct 2009.

[DRLR89] M. Dhome, M. Richetin, J. T. Lapreste, and G. Rives. Determination of the attitude of 3d objects from a single perspective view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(12):1265–1278, Dec 1989.

[DUNI10] B. Drost, M. Ulrich, N. Navab, and S. Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 998–1005, June 2010.

[EMC09] M. Ebrahimi and W. W. Mayol-Cuevas. Susure: Speeded up surround extrema feature detector and descriptor for realtime applications. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 9–14, June 2009.

[EWPA16] Eyþór Rúnar Eiríksson, Jakob Wilm, David Bue Pedersen, and Henrik Aanæs. *Precision and Accuracy Parameters in Structured Light 3-D Scanning*, pages 7–15. 2016. The Archives are open access publications, they are published under the Creative Common Attribution 3.0 License.

[FB81] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[FBF77] Jerome H. Friedman, Jon Louis Bentley, and Raphael Ari Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Softw.*, 3(3):209–226, September 1977.

[FBMN14] L. Ferraz, X. Binefa, and F. Moreno-Noguer. Very fast solution to the pnp problem with algebraic outlier rejection. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 501–508, June 2014.

[FDvdH07] A. Flint, A. Dick, and A. v. d. Hengel. Thrift: Local 3d structure recognition. In *Digital Image Computing Techniques and Applications, 9th Biennial Conference of the Australian Pattern Recognition Society on*, pages 182–188, Dec 2007.

[FDvdH08] A. Flint, A. Dick, and A. van den Hengel. Local 3d structure recognition in range images. *IET Computer Vision*, 2(4):208–217, December 2008.

[FHK+04] Andrea Frome, Daniel Huber, Ravi Kolluri, Thomas Bülow, and Jitendra Malik. *Computer Vision - ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part III*, chapter Recognizing Objects in Range Data Using Regional Point Descriptors, pages 224–237. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[Fio01] P. D. Fiore. Efficient linear solution of exterior orientation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):140–148, Feb 2001.

[FN75] K. Fukunage and P. M. Narendra. A branch and bound algorithm for computing k-nearest neighbors. *IEEE Trans. Comput.*, 24(7):750–753, July 1975.

[FSDK11] R. Furukawa, R. Sagawa, A. Delaunoy, and H. Kawasaki. Multi-view projectors/cameras system for 3d reconstruction of dynamic scenes. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1602–1609, Nov 2011.

[GBS+14] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, and J. Wan. 3d object recognition in cluttered scenes with local surface features: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2270–2287, Nov 2014.

[GBS+16] Yulan Guo, Mohammed Bennamoun, Ferdous Sohel, Min Lu, Jianwei Wan, and Ngai Ming Kwok. A comprehensive performance evaluation of 3d local feature descriptors. *International Journal of Computer Vision*, 116(1):66–89, 2016.

[GC11] Peter Gorle and Andrew Clive. Positive impact of industrial robots on employment. *IFR - International Federation of Robotics*, 2011. Available at `http://www.ifr.org/uploads/media/Metra_Martech_Study_on_robots_02.pdf`.

[GCF12] V. Garro, F. Crosilla, and A. Fusiello. Solving the pnp problem with anisotropic orthogonal procrustes analysis. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization Transmission*, pages 262–269, Oct 2012.

[Gen11] Jason Geng. Structured-light 3d surface imaging: a tutorial. *Adv. Opt. Photon.*, 3(2):128–160, Jun 2011.

[GES+10] Thilo Grundmann, Robert Eidenberger, Martin Schneider, Michael Fiegert, and Georg v Wichert. Robust high precision 6d pose determination in complex environments for robotic manipulation. In *Proc. Workshop Best Practice in 3D Perception and Modeling for Mobile Manipulation at the Int. Conf. Robotics and Automation*, pages 1–6, 2010.

[GHTC03] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):930–943, Aug 2003.

[GIV10] A. Georgopoulos, Ch. Ioannidis, and A. Valanis. Assessing the performance of a structured light scanner. *Int Arch Ph*, 38:250–255, 2010.

[GL06] Iryna Gordon and David G. Lowe. *Toward Category-Level Object Recognition*, chapter What and Where: 3D Object Recognition with Accurate Pose, pages 67–82. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[GMBR10] Arturo Gil, Oscar Martinez Mozos, Monica Ballesta, and Oscar Reinoso. A comparative evaluation of interest point detectors and local descriptors for visual slam. *Mach. Vision Appl.*, 21(6):905–920, October 2010.

[GSB+13a] Y. Guo, F. A. Sohel, M. Bennamoun, J. Wan, and M. Lu. Integrating shape and color cues for textured 3d object recognition. In *2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA)*, pages 1614–1619, June 2013.

[GSB+13b] Yulan Guo, Ferdous Sohel, Mohammed Bennamoun, Min Lu, and Jianwei Wan. Rotational projection statistics for 3d local surface description and object recognition. *International Journal of Computer Vision*, 105(1):63–86, 2013.

[GSB+15] Yulan Guo, Ferdous Sohel, Mohammed Bennamoun, Jianwei Wan, and Min Lu. A novel local surface feature for 3d object recognition under clutter and occlusion. *Information Sciences*, 293:196 – 213, 2015.

[Gue00] Jens Guehring. Dense 3d surface acquisition by structured light using off-the-shelf components. volume 4309, pages 220–231, 2000.

[Har97] R. I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, Jun 1997.

[Har98] R. I. Hartley. Minimizing algebraic error in geometric estimation problems. In *Computer Vision, 1998. Sixth International Conference on*, pages 469–476, Jan 1998.

[HCLL89] R. Horaud, B. Conio, O. Leboulleux, and B. Lacolle. An analytic solution for the perspective 4-point problem. In *Computer Vision and Pattern Recognition, 1989. Proceedings CVPR '89., IEEE Computer Society Conference on*, pages 500–507, Jun 1989.

[HD95] Radu P. Horaud and Fadi Dornaika. Hand-eye calibration. *Int. J. Robot. Res.*, 14(3):195–210, June 1995.

[HDF12] Jared Heinly, Enrique Dunn, and Jan-Michael Frahm. Comparative evaluation of binary features. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part II*, ECCV'12, pages 759–773, Berlin, Heidelberg, 2012. Springer-Verlag.

[HHP16] J. Heller, M. Havlena, and T. Pajdla. Globally optimal hand-eye calibration using branch-and-bound. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(5):1027–1033, May 2016.

[HK97] E. Horn and N. Kiryati. Toward optimal structured light patterns. In *3-D Digital Imaging and Modeling, 1997. Proceedings., International Conference on Recent Advances in*, pages 28–35, May 1997.

[HKH+12] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments. *Int. J. Rob. Res.*, 31(5):647–663, April 2012.

[HLON] Bert M. Haralick, Chung-Nan Lee, Karsten Ottenberg, and Michael Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *International Journal of Computer Vision*, 13(3):331–356.

[Hor87] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987.

[HPS+14] Kent Hansen, Jeppe Pedersen, Thomas Sølund, Henrik Aanæs, and Dirk Kraft. A structured light scanner for hyper flexible industrial automation. In *2014 2nd International Conference on 3D Vision*, volume 1, pages 401–408, Dec 2014. Peer-reviewed.

[HR11] J. A. Hesch and S. I. Roumeliotis. A direct least-squares (dls) method for pnp. In *2011 International Conference on Computer Vision*, pages 383–390, Nov 2011.

[HS88] Chris Harris and Mike Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.

[HS97] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 1106–1112, Jun 1997.

[HW02] Z. Y. Hu and F. C. Wu. A note on the number of solutions of the noncoplanar p4p problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):550–555, Apr 2002.

[HZ04] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[IKHM11] Shahram Izadi, David Kim, Otmar Hilliges, and David Molyneaux. KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. *Proceedings of the 24th*, 2011.

[ISM84] S. Inokuchi, K. Sato, and F. Matsuda. Range-imaging system for 3-d object recognition. 1984.

[JBK15] Troels Bo Jørgensen, Anders Glent Buch, and Dirk Kraft. *Geometric Edge Description and Classification in Point Cloud Data with Application to 3D Object Recognition*, volume 1, pages 333–340. Institute for Systems and Technologies of Information, Control and Communication, Portugal, 3 2015.

[JDV+14] R. Jensen, A. Dahl, G. Vogiatzis, E. Tola, and H. Aanæs. Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413, June 2014.

[JG09] Luo Juan and Oubong Gwon. A Comparison of SIFT, PCA-SIFT and SURF. *International Journal of Image Processing (IJIP)*, 3(4):143–152, 2009.

[JH99] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449, May 1999.

[KE12] Kourosh Khoshelham and Sander Oude Elberink. Accuracy and resolution of Kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–54, January 2012.

[KF14] L. Kneip and P. Furgale. Opengv: A unified and generalized approach to real-time calibrated geometric vision. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8, May 2014.

[KFS13] L. Kneip, P. Furgale, and R. Siegwart. Using multi-camera systems in robotics: Efficient solutions to the npnp problem. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3770–3776, May 2013.

[KHRF11] Michael Krainin, Peter Henry, Xiaofeng Ren, and Dieter Fox. Manipulator and object tracking for in-hand 3d object modeling. *Int. J. Rob. Res.*, 30(11):1311–1327, September 2011.

[KM07] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, ISMAR '07, pages 1–10, Washington, DC, USA, 2007. IEEE Computer Society.

[Kon10] Kurt Konolige. Projected Texture Stereo. In *IEEE International Conference on Robotics and Automation, ICRA*, 2010.

[Kos96] Andreas Koschan. Color stereo vision using hierarchical block matching and active color illumination. In *Pattern Recognition*, volume I, pages 835–839, 1996.

[KP06] Akash Kushal and Jean Ponce. *Computer Vision – ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part II*, chapter Modeling 3D Objects from Stereo Views and Recognizing Them in Photographs, pages 563–574. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[KS04] Yan Ke and R. Sukthankar. Pca-sift: a more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–506–II–513 Vol.2, June 2004.

[KSS11] L. Kneip, D. Scaramuzza, and R. Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 2969–2976, Washington, DC, USA, 2011. IEEE Computer Society.

[KTF11] B. Kaneva, A. Torralba, and W. T. Freeman. Evaluation of image features using a photorealistic virtual world. In *2011 International Conference on Computer Vision*, pages 2282–2289, Nov 2011.

[KWH13] H. Kagermann, W. Wahlster, and J. Helbig. Final report of the industrie 4.0 working group-acatech: Recommendations for implementing the strategic initiative industrie 4.0. pages 1–82, April 2013.

[KWZK95] Sing Bing Kang, Jon A. Webb, C. Lawrence Zitnick, and Takeo Kanade. A multibaseline stereo system with active illumination and real-time image acquisition. In *International Conference on Computer Vision (ICCV 95)*, 1995.

[KZB04] T. Kadir, A. Zisserman, and J. M. Brady. An affine invariant salient region detector. In *European Conference on Computer Vision*. Springer-Verlag, 2004.

[LCS11] S. Leutenegger, M. Chli, and R. Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *2011 International Conference on Computer Vision*, pages 2548–2555, Nov 2011.

[LFNP09] V. Lepetit, F.Moreno-Noguer, and P.Fua. Epnp: An accurate o(n) solution to the pnp problem. *International Journal Computer Vision*, 81(2), 2009.

[LH15] Gil Levi and Tal Hassner. LATCH: learned arrangements of three patch codes. *CoRR*, abs/1501.03719, 2015.

[LHM00] C. P. Lu, G. D. Hager, and E. Mjolsness. Fast and globally convergent pose estimation from video images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):610–622, Jun 2000.

[Li14] Li Li. Image matching algorithm based on feature-point and daisy descriptor. *Journal of Multimedia*, 9(6), 2014.

[Lim09] Jongwoo Lim. Optimized projection pattern supplementing stereo systems. In *IEEE International Conference on Robotics and Automation*, pages 2823–2829. Ieee, May 2009.

[Lin98] Tony Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116, 1998.

[LMYG04] Ting Liu, Andrew W. Moore, Ke Yang, and Alexander G. Gray. An investigation of practical approximate nearest neighbor algorithms. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 825–832. MIT Press, Cambridge, MA, 2004.

[Low87] D G Lowe. Three-dimensional object recognition from single two-dimensional images. *Artif. Intell.*, 31(3):355–395, March 1987.

[Low99] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150–1157 vol.2, 1999.

[Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

[LSP05] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1265–1278, Aug 2005.

[LWTD15] Yali Li, Shengjin Wang, Qi Tian, and Xiaoqing Ding. A survey of recent advances in visual feature detection. *Neurocomputing*, 149(PB):736–751, 2 2015.

[LXX12] S. Li, C. Xu, and M. Xie. A robust o(n) solution to the perspective-n-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1444–1450, July 2012.

[MBAG10] D. Marimon, A. Bonnin, T. Adamek, and R. Gimeno. Darts: Efficient scale-space extraction of daisy keypoints. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2416–2423, June 2010.

[MBO06] A.S. Mian, M. Bennamoun, and R. Owens. Three-dimensional model-based object recognition and segmentation in cluttered scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(10):1584–1601, Oct 2006.

[MBO10] A. Mian, M. Bennamoun, and R. Owens. On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. *International Journal of Computer Vision*, 89(2-3):348–361, 2010.

[MCUP02] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, pages 36.1–36.10. BMVA Press, 2002. doi:10.5244/C.16.36.

[MHB+10] Elmar Mair, Gregory D. Hager, Darius Burschka, Michael Suppa, and Gerhard Hirzinger. Adaptive and generic corner detection based on the accelerated segment test. In *Proceedings of the European Conference on Computer Vision (ECCV'10)*, September 2010.

[MJWW15] Dibyendu Mukherjee, Q. M. Jonathan Wu, and Guanghui Wang. A comparative experimental study of image feature detectors and descriptors. *Mach. Vision Appl.*, 26(4):443–466, May 2015.

[ML09] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *In VISAPP International Conference on Computer Vision Theory and Applications*, pages 331–340, 2009.

[ML12] Marius Muja and David G. Lowe. Fast matching of binary features. In *Proceedings of the 2012 Ninth Conference on Computer and Robot Vision*, CRV '12, pages 404–410, Washington, DC, USA, 2012. IEEE Computer Society.

[ML14] M. Muja and D. G. Lowe. Scalable nearest neighbor algorithms for high dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2227–2240, Nov 2014.

[Moo00] Andrew W. Moore. The anchors hierarchy: Using the triangle inequality to survive high dimensional data. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*,

UAI'00, pages 397–405, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[MP05] P. Moreels and P. Perona. Evaluation of features detectors and descriptors based on 3d objects. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 800–807 Vol. 1, Oct 2005.

[MS04] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *Int. J. Comput. Vision*, 60(1):63–86, October 2004.

[MS05] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, Oct 2005.

[MSC05] E. Marchand, F. Spindler, and F. Chaumette. Visp for visual servoing: a generic software platform with a wide class of robot control skills. *IEEE Robotics and Automation Magazine*, 12(4):40–52, December 2005.

[MTS+05] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *Int. J. Comput. Vision*, 65(1-2):43–72, November 2005.

[Mun06] Joseph L. Mundy. Object recognition in the geometric era: A retrospective. In *Toward Category-Level Object Recognition*, pages 3–28, 2006.

[NF91] D. K. Naidu and R. B. Fisher. *A Comparative Analysis of Algorithms for Determining the Peak Position of a Stripe to Sub-pixel Accuracy*, pages 217–225. Springer London, London, 1991.

[NIH11] RA Newcombe, S Izadi, and O Hilliges. KinectFusion: Real-time dense surface mapping and tracking. *(ISMAR), 2011 10th*, 2011.

[Nis84] HK Nishihara. Prism, a practical real-time imaging stereo matcher, ai memo 780. Technical report, 1984.

[NKM10] Takahiro Nakada, Satoshi Kagami, and Hiroshi Mizoguchi. Sift-cloud-model for object detection and pose estimation with GPGPU acceleration. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, October 18-22, 2010, Taipei, Taiwan*, pages 1748–1753, 2010.

[NN94] Shree K. Nayar and Yasuo Nakagawa. Shape from focus. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(8):824–831, 1994.

[NS06] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2161–2168, 2006.

[ODD96] Denis Oberkampf, Daniel F. DeMenthon, and Larry S. Davis. Iterative pose estimation using coplanar feature points. *Computer Vision and Image Understanding*, 63(3):495 – 511, 1996.

[OKO09] C. Olsson, F. Kahl, and M. Oskarsson. Branch-and-bound methods for euclidean registration problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):783–794, May 2009.

[PA82] JL Posdamer and MD Altschuler. Surface measurement by space-encoded projected beam systems. In *Computer graphics and image processing*, pages 1–17, 1982.

[PAGIoT13] Adrien Bartoli Pablo Alcantarilla (Georgia Institute of Technolog), Jesus Nuevo (TrueVision Solutions AU). Fast explicit diffusion for accelerated features in nonlinear scale spaces. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2013.

[PHB11] Dejan Pangercic, Vladimir Haltakov, and Michael Beetz. Fast and robust object detection in household environments using vocabulary trees with sift descriptors. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Workshop on Active Semantic Perception and Object Search in the Real World*, San Francisco, CA, USA, September, 25–30 2011.

[PK11] Nicolas Pugeault and Norbert Krüger. Temporal accumulation of oriented visual features. *J. Vis. Comun. Image Represent.*, 22(2):153–163, February 2011.

[PLW+08] Giorgio Panin, Claus Lenz, Martin Wojtczyk, Suraj Nair, Erwin Roth, Thomas Friedlhuber, and Alois Knoll. A unifying software architecture for model-based visual tracking. volume 6813, pages 681303–681303–14, 2008.

[PM94] F. C. Park and B. J. Martin. Robot sensor calibration: solving ax=xb on the euclidean group. *IEEE Transactions on Robotics and Automation*, 10(5):717–721, Oct 1994.

[PMC+11] Francois Pomerleau, Stephane Magnenat, Francis Colas, Ming Liu, and Roland Siegwart. Tracking a depth camera: Parameter exploration for fast ICP. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3824–3829. Ieee, September 2011.

[PS11] A. Petrelli and L. Di Stefano. On the repeatability of the local reference frame for partial shape matching. In *2011 International Conference on Computer Vision*, pages 2244–2251, Nov 2011.

[PTB+15] J. Posada, C. Toro, I. Barandiaran, D. Oyarzun, D. Stricker, R. de Amicis, E. B. Pinto, P. Eisert, J. Döllner, and I. Vallarino. Visual computing as a key enabling technology for industrie 4.0 and industrial internet. *IEEE Computer Graphics and Applications*, 35(2):26–40, Mar 2015.

[PZC13] Giuliano Pasqualotto, Pietro Zanuttigh, and Guido M. Cortelazzo. Combining color and shape descriptors for 3d model retrieval. *Signal Processing: Image Communication*, 28(6):608 – 623, 2013.

[QL99] Long Quan and Zhongdan Lan. Linear n-point camera pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):774–780, Aug 1999.

[QLZ10] Wang Qi, Fu Li, and Liu Zhenzhong. Review on camera calibration. In *2010 Chinese Control and Decision Conference*, pages 3354–3358, May 2010.

[Qui86] J. R. Quinlan. Induction of decision trees. *Mach. Learn.*, 1(1):81–106, March 1986.

[RBB09] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 3212–3217, May 2009.

[RBMB08] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz. Aligning point cloud views using persistent feature histograms. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3384–3391, Sept 2008.

[RCSM01] S. Ruiz-Correa, L. G. Shapiro, and M. Melia. A new signature-based method for efficient 3-d object recognition. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–769–I–776 vol.1, 2001.

[RD06] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Proceedings of the 9th European Conference on Computer Vision - Volume Part I*, ECCV'06, pages 430–443, Berlin, Heidelberg, 2006. Springer-Verlag.

[RDLD97] S. Remy, M. Dhome, J. M. Lavest, and N. Daucher. Hand-eye calibration. In *Intelligent Robots and Systems, 1997. IROS '97., Proceedings of the 1997 IEEE/RSJ International Conference on*, volume 2, pages 1057–1065 vol.2, Sep 1997.

[RL01] Szymon Rusinkiewicz and Marc Levoy. Efcient Variants of the ICP Algorithm. In *INTERNATIONAL CONFERENCE ON 3-D DIGITAL IMAGING AND MODELING*, 2001.

[RLSP06] Fred Rothganger, Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *International Journal of Computer Vision*, 66(3):231–259, 2006.

[Rob63] LG Roberts. *Machine perception of three-dimensional solids.* PhD thesis, 1963.

[RRKB11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571, Nov 2011.

[RW08] Peter M. Roth and Martin Winter. Survey of appearance-based methods for object recognition. tech. rep., 2008.

[SA89] Y. C. Shiu and S. Ahmad. Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form ax=xb. *IEEE Transactions on Robotics and Automation*, 5(1):16–29, Feb 1989.

[SAB02] Joaquim Salvi, Xavier Armangué, and Joan Batlle. A comparative review of camera calibrating methods with accuracy evaluation. *Pattern Recognition*, 35(7):1617 – 1635, 2002.

[SABA13] Thomas Sølund, Rasmus Hasle Andersen, Anders Billesø Beck, and Henrik Aanæs. Combining 3D Object Modelling and Robot Skills for Intuitive Instruction of Robotic co-workers. In *2nd AAU Workshop on Robotics*, 2013. Peer-reviewed.

[SAH08] C. Silpa-Anan and R. Hartley. Optimised kd-trees for fast image descriptor matching. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[SB97] Stephen M. Smith and J. Michael Brady. Susan—a new approach to low level image processing. *Int. J. Comput. Vision*, 23(1):45–78, May 1997.

[SBKA16] Thomas Sølund, Anders G. Buch, Norbert Krüger, and Henrik Aanæs. A large-scale 3d object recognition dataset. In *2016 4th International Conference on 3D Vision*, Submitted and under double-blind review. Paper notification date is the 31th of August 2016.

[SBM98] J. Salvi, J. Batlle, and E. Mouaddib. A robust-coded pattern projection for dynamic 3d scene measurement. *Pattern Recognition Letters*, 19(11):1055 – 1065, 1998.

[SCD+06] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 519–528, June 2006.

[Sch]

[SD16] S. Sharma and S. D'Amico. Comparative assessment of techniques for initial pose estimation using monocular vision. *Acta Astronautica*, 123:435–445, June 2016.

[SEH12] Mili Shah, Roger D. Eastman, and Tsai Hong. An overview of robot-sensor calibration methods for evaluation of perception systems. In *Proceedings of the Workshop on Performance Metrics for Intelligent Systems*, PerMIS '12, pages 15–20, New York, NY, USA, 2012. ACM.

[SFK14] R. Sagawa, R. Furukawa, and H. Kawasaki. Dense 3d reconstruction from high frame-rate video using a static grid pattern. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(9):1733–1747, Sept 2014.

[SFPL10] Joaquim Salvi, Sergio Fernandez, Tomislav Pribanic, and Xavier Llado. A state of the art in structured light patterns for surface profilometry. *Pattern Recogn.*, 43(8):2666–2680, August 2010.

[SH06] K. H. Strobl and G. Hirzinger. Optimal hand-eye calibration. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4647–4653, Oct 2006.

[SHK+] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nesic, Xi Wang, and Porter Westling. In *GCPR*.

[SKKF11] R. Sagawa, H. Kawasaki, S. Kiyota, and R. Furukawa. Dense one-shot 3d reconstruction by detecting continuous regions with parallel line projection. In *2011 International Conference on Computer Vision*, pages 1911–1918, Nov 2011.

[SKSB10] Jurgen Sturm, Kurt Konolige, Cyrill Stachniss, and Wolfram Burgard. Vision-based detection for learning articulation models of cabinet doors and drawers in household environments. *2010 IEEE International Conference on Robotics and Automation*, pages 362–368, May 2010.

[SLK15] Hamed Sarbolandi, Damien Lefloch, and Andreas Kolb. Kinect range sensing: Structured-light versus time-of-flight kinect. *Computer Vision and Image Understanding*, 139:1 – 20, 2015.

[SMB00] Cordelia Schmid, Roger Mohr, and Christian Bauckhage. Evaluation of interest point detectors. *Int. J. Comput. Vision*, 37(2):151–172, June 2000.

[SP08] Gerald Schweighofer and Axel Pinz. Globally optimal o(n) solution to the pnp problem for general camera models. In *Proceedings of the British Machine Vision Conference 2008, Leeds, September 2008*, pages 1–10, 2008.

[SRT+11] Peter Sturm, Srikumar Ramalingam, Jean-Philippe Tardif, Simone Gasparini, and João Barreto. Camera models and fundamental concepts used in geometric computer vision. *Found. Trends. Comput. Graph. Vis.*, 6(1–2):1–183, January 2011.

[SS03] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–195–I–202 vol.1, June 2003.

[SSB+ed] Thomas Sølund, Thiusius Rajeeth Savarimuthu, Anders Glent Buch, Anders Billesø Beck, Norbert Krüger, and Henrik Aanæs. Teach it yourself - fast modeling of industrial objects for 6d pose estimation. In *Computer Vision Systems - 10th International Conference, ICVS 2015, Copenhagen, Denmark, July 6-9, 2015, Proceedings*, pages 289–302, 2015, Peer-reviewed.

[ST94] Jianbo Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pages 593–600, Jun 1994.

[Ste99] Charles V. Stewart. Robust parameter estimation in computer vision. *SIAM Rev.*, 41(3):513–537, September 1999.

[STS14] Samuele Salti, Federico Tombari, and Luigi Di Stefano. Shot: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125(0):251 – 264, 2014.

[SWW13] Federico Sukno, John Waddington, and Paul F. Whelan. Rotationally invariant 3d shape contexts using asymmetry patterns. In *GRAPP & IVAPP 2013: Proceedings of the International Conference on Computer Graphics Theory and Applications and International Conference on Information Visualization Theory and Applications, Barcelona, Spain, 21-24 February, 2013.*, pages 7–17, 2013.

[TB13] Moritz Tenorth and Michael Beetz. Knowrob: A knowledge processing infrastructure for cognition-enabled robots. *The International Journal of Robotics Research*, 32(5):566–590, 2013.

[TBJG07] B. Taati, M. Bondy, P. Jasiobedzki, and M. Greenspan. Variable dimensional local shape descriptors for object recognition in range data. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct 2007.

[TK91] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. Technical report, International Journal of Computer Vision, 1991.

[TKM11] H. Tsubota, S. Kagami, and H. Mizoguchi. Sift-cloud-model generation method for 6d pose estimation and its evaluation. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, pages 3323–3328, Oct 2011.

[TL89] R. Y. Tsai and R. K. Lenz. A new technique for fully autonomous and efficient 3d robotics hand/eye calibration. *IEEE Transactions on Robotics and Automation*, 5(3):345–358, Jun 1989.

[TL12] T. Trzcinski and V. Lepetit. Efficient Discriminative Projections for Compact Binary Descriptors. In *European Conference on Computer Vision*, 2012.

[TLF10] E. Tola, V. Lepetit, and P. Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):815–830, May 2010.

[TM08] Tinne Tuytelaars and Krystian Mikolajczyk. Local invariant feature detectors: A survey. *Found. Trends. Comput. Graph. Vis.*, 3(3):177–280, July 2008.

[Tri99] B. Triggs. Camera pose and calibration from 4 or 5 known 3d points. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 278–284 vol.1, 1999.

[Tsa86] R. Y. Tsai. An efficient and accurate camera calibration technique for 3D machine vision. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 364–374, Miami, June 1986.

[TSDS10] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *Proceedings of the 11th European Conference on Computer Vision Conference on Computer Vision: Part III*, ECCV'10, pages 356–369, Berlin, Heidelberg, 2010. Springer-Verlag.

[TSS11] F. Tombari, S. Salti, and L. Di Stefano. A combined texture-shape descriptor for enhanced 3d feature matching. In *2011 18th IEEE International Conference on Image Processing*, pages 809–812, Sept 2011.

[TV04] J. W. H. Tangelder and R. C. Veltkamp. A survey of content based 3d shape retrieval methods. In *Shape Modeling Applications, 2004. Proceedings*, pages 145–156, June 2004.

[TVG04] Tinne Tuytelaars and Luc Van Gool. Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*, 59(1):61–85, 2004.

[Ume91] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, Apr 1991.

[UWS09] M. Ulrich, C. Wiedemann, and C. Steger. Cad-based recognition of 3d objects in monocular images. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 1191–1198, May 2009.

[VFJM09] F. Viksten, P. E. Forssen, B. Johansson, and A. Moe. Comparison of local image descriptors for full 6 degree-of-freedom pose estimation. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 2779–2786, May 2009.

[VSNP06] F. Viksten, R. Soderberg, K. Nordberg, and C. Perwass. Increasing pose estimation performance using multi-cue integration. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 3760–3767, May 2006.

[Wan92] C. C. Wang. Extrinsic calibration of a vision sensor mounted on a robot. *IEEE Transactions on Robotics and Automation*, 8(2):161–175, Apr 1992.

[WBD+11] Markus Waibel, Michael Beetz, Raffaello D'Andrea, Rob Janssen, Moritz Tenorth, Javier Civera, Jos Elfring, Dorian Gálvez-López, Kai Häussermann, J.M.M. Montiel, Alexander Perzylo, Björn Schießle, Oliver Zweigle, and René van de Molengraft. RoboEarth - A World Wide Web for Robots. *Robotics & Automation Magazine*, 18(2):69–82, 2011.

[WBG+12] M.J. Westoby, J. Brasington, N.F. Glasser, M.J. Hambrey, and J.M. Reynolds. 'structure-from-motion' photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology*, 179:300 – 314, 2012.

[WHB09] S. Winder, Gang Hua, and M. Brown. Picking the best daisy. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 0:178–185, 2009.

[WIS+10] K. Welke, J. Issac, D. Schiebener, T. Asfour, and R. Dillmann. Autonomous acquisition of visual multi-view object representations for object recognition on a humanoid robot. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2012–2019, May 2010.

[Wit84] A. Witkin. Scale-space filtering: A new approach to multi-scale description. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '84.*, volume 9, pages 150–153, Mar 1984.

[WMSM91] W. J. Wolfe, D. Mathis, C. W. Sklair, and M. Magee. The perspective view of three points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(1):66–73, Jan 1991.

[WOL14] J. Wilm, O. V. Olesen, and R. Larsen. Slstudio: Open-source framework for real-time structured light. In *2014 4th International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–4, Oct 2014.

[WSRK11] M. Weinmann, C. Schwartz, R. Ruiters, and R. Klein. A multi-camera, multi-projector super-resolution framework for structured light. In *2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pages 397–404, May 2011.

[YC14] Xin Yang and Kwang-Ting Cheng. Local difference binary for ultrafast and distinctive feature description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1):188–194, 2014.

[YF02] S. M. Yamany and A. A. Farag. Surface signatures: an orientation independent free-form surface representation scheme for the purpose of objects registration and matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1105–1120, Aug 2002.

[Yia93] Peter N. Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '93, pages 311–321, Philadelphia, PA, USA, 1993. Society for Industrial and Applied Mathematics.

[ZBVH09] Andrei Zaharescu, Edmond Boyer, Kiran Varanasi, and Radu P. Horaud. Surface feature detection and description with applications to mesh matching. In *International Conference on Computer Vision and Pattern Recognition, CVPR'09, June, 2009*, pages 373–380, Miami, Etats-Unis, June 2009. IEEE.

[ZCS] Li Zhang, Brian Curless, and Steven M. Seitz. In *CVPR (2)*.

[Zha00] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334, November 2000.

[Zha10] Song Zhang. Recent progresses on real-time 3d shape measurement using digital fringe projection techniques. *Optics and Lasers in Engineering*, 48(2):149 – 158, 2010. Fringe Projection Techniques.

[Zho09] Y. Zhong. Intrinsic shape signatures: A shape descriptor for 3d object recognition. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 689–696, Sept 2009.

[ZKS+13] Y. Zheng, Y. Kuang, S. Sugimoto, K. Åström, and M. Okutomi. Revisiting the pnp problem: A fast, general and optimal solution. In *2013 IEEE International Conference on Computer Vision*, pages 2344–2351, Dec 2013.

[ZPMV11] M. Zillich, J. Prankl, T. Mörwald, and M. Vincze. Knowing your limits - self-evaluation and prediction in object recognition. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 813–820, Sept 2011.

[ZRS94] Hanqi Zhuang, Z. S. Roth, and R. Sudhakar. Simultaneous robot/world and tool/flange calibration by solving homogeneous transformation equations of the form ax=yb. *IEEE Transactions on Robotics and Automation*, 10(4):549–554, Aug 1994.

[ZS93] Hanqi Zuang and Yiu Cheung Shiu. A noise-tolerant algorithm for robotic hand-eye calibration with or without sensor orientation measurement. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(4):1168–1175, Jul 1993.

[ZSO13] Yinqiang Zheng, Shigeki Sugimoto, and Masatoshi Okutomi. Aspnp: An accurate and scalable solution to the perspective-n-point problem. *IEICE Transactions*, 96-D(7):1525–1535, 2013.

[ZTCS99] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape from shading: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(8):690–706, August 1999.

[ZYH+13] Xin Zhang, Yee-Hong Yang, Zhiguang Han, Hui Wang, and Chao Gao. Object class detection: A survey. *ACM Comput. Surv.*, 46(1):10:1–10:53, July 2013.